A Novel Combination of Answer Set Programming with Description Logics for the Semantic Web

Thomas Lukasiewicz *

Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Rome, Italy
lukasiewicz@dis.uniroma1.it

Abstract. We present a novel combination of disjunctive logic programs under the answer set semantics with description logics for the Semantic Web. The combination is based on a well-balanced interface between disjunctive logic programs and description logics, which guarantees the decidability of the resulting formalism without assuming syntactic restrictions. We show that the new formalism has very nice semantic properties. In particular, it faithfully extends both disjunctive programs and description logics. Furthermore, we describe algorithms for reasoning in the new formalism, and we give a precise picture of its computational complexity. We also provide a special case with polynomial data complexity.

1 Introduction

The *Semantic Web* [4,14] aims at an extension of the current World Wide Web by standards and technologies that help machines to understand the information on the Web so that they can support richer discovery, data integration, navigation, and automation of tasks. The main ideas behind it are to add a machine-readable meaning to Web pages, to use ontologies for a precise definition of shared terms in Web resources, to use knowledge representation technology for automated reasoning from Web resources, and to apply cooperative agent technology for processing the information of the Web.

The Semantic Web consists of several hierarchical layers, where the *Ontology layer*, in form of the *OWL Web Ontology Language* [41,22], is currently the highest layer of sufficient maturity. OWL consists of three increasingly expressive sublanguages, namely *OWL Lite*, *OWL DL*, and *OWL Full*. OWL Lite and OWL DL are essentially very expressive description logics with an RDF syntax [22]. As shown in [19], ontology entailment in OWL Lite (resp., OWL DL) reduces to knowledge base (un)satisfiability in the description logic $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$). As a next important step in the development of the Semantic Web, one aims at sophisticated representation and reasoning capabilities for the *Rules*, *Logic*, and *Proof layers* of the Semantic Web.

In particular, there is a large body of work on integrating rules and ontologies, which is a key requirement of the layered architecture of the Semantic Web. Significant research efforts focus on hybrid integrations of rules and ontologies, called *description logic programs*[1] (or *dl-programs*), which are of the form $KB = (L, P)$, where L is a

* Alternate address: Institut für Informationssysteme, Technische Universität Wien, Favoritenstraße 9-11, A-1040 Vienna, Austria; e-mail: lukasiewicz@kr.tuwien.ac.at.

[1] Note that we use the notion of "description logic programs" in a generic way, that is, to denote a class of different formalisms, similarly to the notion of "description logics".

E. Franconi, M. Kifer, and W. May (Eds.): ESWC 2007, LNCS 4519, pp. 384–398, 2007.

description logic knowledge base and P is a finite set of rules involving either queries to L in a loose integration (see especially [11,12,9,10]) or concepts and roles from L as unary resp. binary predicates in a tight integration (see especially [36,37]).

However, especially the tight integration of rules and ontologies presents many semantic and computational difficulties [37]. Since many expressive description logics are very close to the decidability / undecidability frontier (such as $\mathcal{SHOIN}(\mathbf{D})$, which is only decidable when number restrictions are limited to simple abstract roles [23]), developing decidable extensions of them by rules turns out to be a naturally hard task, and often comes along with strong syntactic restrictions on the resulting language (such as syntactic safety conditions and/or syntactic partitionings of the vocabulary).

Nevertheless, in rule-based systems in the Semantic Web, we would like to use vocabulary from formal ontologies, and we would like to do it without syntactic restrictions. In this paper, we show that the main difficulties with the above tight integrations of rules and ontologies lies actually in the *perspective of the integration*. That is, they all look from the *perspective of description logics* at the integration of rules and ontologies. However, for extending certain kinds of rule-based systems by vocabulary from formal ontologies, we actually do not need the full power of a rule-based extension of description logics. This is the main idea behind this paper. More precisely, we look at the integration of rules and ontologies from the *perspective of rule-based systems*.

The main contributions of this paper can be summarized as follows:

- We present a new combination of disjunctive logic programs under the answer set semantics with description logics. In detail, we present a new form of tightly integrated disjunctive dl-programs $KB = (L, P)$ under the answer set semantics, which allows for decidable reasoning, without assuming any syntactic restrictions (see Section 8 for a detailed comparison to previous approaches to dl-programs).
- Intuitively, the main idea behind the semantics of such dl-programs $KB = (L, P)$ is to interpret P relative to Herbrand interpretations that also satisfy L, while L is interpreted relative to general interpretations over a first-order domain. That is, we modularly combine the standard semantics of disjunctive programs P and of description logics L, via a well-balanced interface between P and L.
- We show that the new approach to disjunctive dl-programs under the answer set semantics has very nice semantic features. In particular, the cautious answer set semantics faithfully extends both disjunctive programs and description logics, and its closed-world property is limited to explicit default-negated atoms in rule bodies. Furthermore, the new approach does not need the unique name assumption.
- We also analyze the computational aspects of the new formalism. We describe algorithms for deciding answer set existence, brave consequences, and cautious consequences. This shows in particular that these decision problems are all decidable. We also draw a precise picture of the complexity of all these decision problems.
- Finally, we delineate a special case of stratified normal dl-programs where the above decision problems all have a polynomial data complexity.

The rest of this paper is organized as follows. Sections 2 and 3 recall disjunctive programs under the answer set semantics resp. the description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$. In Section 4, we introduce our novel approach to disjunctive dl-programs under the answer set semantics, and in Section 5, we analyze its semantic properties.

Sections 6 and 7 focus on the computational properties. In Section 8, we discuss related work in the literature. Section 9 summarizes our main results and gives an outlook on future research. Some selected proofs are given in the appendix. Note that detailed proofs of all results are given in the extended report [31].

2 Disjunctive Programs Under the Answer Set Semantics

In this section, we recall disjunctive programs (with default negation) under the answer set semantics; see especially [26] for further details and background.

Syntax. Let Φ be a first-order vocabulary with nonempty finite sets of constant and predicate symbols, but no function symbols. Let $\mathcal{X}$ be a set of variables. A *term* is either a variable from $\mathcal{X}$ or a constant symbol from Φ. An *atom* is of the form $p(t_1, \ldots, t_n)$, where p is a predicate symbol of arity $n \geqslant 0$ from Φ, and $t_1, \ldots, t_n$ are terms. A *literal* l is an atom p or a negated atom $not\, p$. A *disjunctive rule* (or simply *rule*) r is of the form

$$\alpha_1 \vee \cdots \vee \alpha_k \leftarrow \beta_1, \ldots, \beta_n, not\, \beta_{n+1}, \ldots, not\, \beta_{n+m}\,, \tag{1}$$

where $\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_{n+m}$ are atoms and $k, m, n \geqslant 0$. We call $\alpha_1 \vee \cdots \vee \alpha_k$ the *head* of r, while the conjunction $\beta_1, \ldots, \beta_n, not\, \beta_{n+1}, \ldots, not\, \beta_{n+m}$ is its *body*. We define $H(r) = \{\alpha_1, \ldots, \alpha_k\}$ and $B(r) = B^+(r) \cup B^-(r)$, where $B^+(r) = \{\beta_1, \ldots, \beta_n\}$ and $B^-(r) = \{\beta_{n+1}, \ldots, \beta_{n+m}\}$. A *disjunctive program* P is a finite set of disjunctive rules of the form (1). We say P is *positive* iff $m = 0$ for all disjunctive rules (1) in P. We say P is a *normal program* iff $k \leqslant 1$ for all disjunctive rules (1) in P.

Example 2.1. An online store (such as *amazon.com*) may use the subsequent set of rules P to express that (1) pc_1 and pc_2 are personal computers, and obj_3 is either a personal computer or a laptop, (2) pc_1 and obj_3 are brand new, (3) *dell* is the vendor of pc_1 and pc_2, (4) a customer avoids all cameras not on offer, (5) all electronic products that are not brand new are on offer, (6) each vendor of a product is a provider, (7) each entity providing a product is a provider, (8) all related products are similar, and (9) the binary similarity relation on products is transitively closed.

(1) $pc(pc_1);\ \ pc(pc_2);\ \ pc(obj_3) \vee laptop(obj_3);$
(2) $brand\_new(pc_1);\ \ brand\_new(obj_3);$
(3) $vendor(dell, pc_1);\ \ vendor(dell, pc_2);$
(4) $avoid(X) \leftarrow camera(X), not\, offer(X);$
(5) $offer(X) \leftarrow electronics(X), not\, brand\_new(X);$
(6) $provider(V) \leftarrow vendor(V, X), product(X);$
(7) $provider(V) \leftarrow provides(V, X), product(X);$
(8) $similar(X, Y) \leftarrow related(X, Y);$
(9) $similar(X, Z) \leftarrow similar(X, Y), similar(Y, Z).$

Semantics. The answer set semantics of disjunctive programs is defined in terms of finite sets of ground atoms, which represent Herbrand interpretations. Positive disjunctive programs are associated with all their minimal satisfying sets of ground atoms, while the semantics of general disjunctive programs is defined by reduction to the minimal model semantics of positive disjunctive programs via the Gelfond-Lifschitz reduct [15].

More concretely, the *Herbrand universe* of a disjunctive program P, denoted HU_P, is the set of all constant symbols appearing in P. If there is no such constant symbol, then $HU_P = \{c\}$, where c is an arbitrary constant symbol from Φ. As usual, terms, atoms, literals, rules, programs, etc. are *ground* iff they do not contain any variables. The *Herbrand base* of a disjunctive program P, denoted HB_P, is the set of all ground atoms that can be constructed from the predicate symbols appearing in P and the constant symbols in HU_P. Hence, in the standard answer set semantics, the Herbrand base is constructed from all constant and predicate symbols in a given disjunctive program, and thus the Herbrand base is finite. A *ground instance* of a rule $r \in P$ is obtained from r by replacing every variable that occurs in r by a constant symbol from HU_P. We denote by $ground(P)$ the set of all ground instances of rules in P.

An *interpretation* I relative to a disjunctive program P is a subset of HB_P. Informally, every such I represents the Herbrand interpretation in which all $a \in I$ (resp., $a \in HB_P - I$) are true (resp., false). An interpretation I is a *model* of a ground atom $a \in HB_P$, or I *satisfies* a, denoted $I \models a$, iff $a \in I$. We say I is a *model* of a ground rule r, denoted $I \models r$, iff $I \models \alpha$ for some $\alpha \in H(r)$ whenever $I \models B(r)$, that is, $I \models \beta$ for all $\beta \in B^+(r)$ and $I \not\models \beta$ for all $\beta \in B^-(r)$. We say I is a *model* of a disjunctive program P, denoted $I \models P$, iff $I \models r$ for every $r \in ground(P)$. An *answer set* of a positive disjunctive program P is a minimal model of P relative to set inclusion. The *Gelfond-Lifschitz reduct* of a disjunctive program P relative to $I \subseteq HB_P$, denoted P^I, is the ground positive disjunctive program obtained from $ground(P)$ by (i) deleting every rule r such that $B^-(r) \cap I \neq \emptyset$, and (ii) deleting the negative body from each remaining rule. An *answer set* of a disjunctive program P is an interpretation $I \subseteq HB_P$ such that I is an answer set of P^I. A disjunctive program P is *consistent* iff P has an answer set.

Hence, under the answer set semantics, every disjunctive program P is interpreted as its grounding $ground(P)$. Note that the answer sets of any disjunctive program P are also minimal models of P. An equivalent definition of the answer set semantics is based on the so-called *FLP-reduct* [13]: The *FLP-reduct* of a disjunctive program P relative to $I \subseteq HB_P$, denoted P^I, is the set of all $r \in ground(P)$ such that $I \models B(r)$. An interpretation $I \subseteq HB_P$ is an *answer set* of P iff I is a minimal model of P^I.

We finally recall the notions of *cautious* (resp., *brave*) *reasoning* from disjunctive programs under the answer set semantics. A ground atom $a \in HB_P$ is a *cautious* (resp., *brave*) *consequence* of a disjunctive program P under the answer set semantics iff every (resp., some) answer set of P satisfies a. Observe that for positive disjunctive programs P, since the set of all answer sets of P is given by the set of all minimal models of P, it holds that $a \in HB_P$ is a cautious consequence of P under the answer set semantics iff a is a logical consequence of the propositional positive disjunctive program $ground(P)$. Note that, more generally, this result holds also when a is a ground formula constructed from HB_Φ using the Boolean operators $\wedge$ and $\vee$. This means that the *closed-world property* (that is, the derivation of negative facts from the absence of derivations of positive facts) of the above notion of cautious reasoning under the answer set semantics is actually limited to the occurrences of default negations in rule bodies.

3 Description Logics

In this section, we recall the description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, which stand behind the web ontology languages OWL Lite and OWL DL [19], respectively. Intuitively, description logics model a domain of interest in terms of concepts and roles, which represent classes of individuals and binary relations between classes of individuals, respectively. A description logic knowledge base encodes especially subset relationships between concepts, subset relationships between roles, the membership of individuals to concepts, and the membership of pairs of individuals to roles.

Syntax. We first describe the syntax of $\mathcal{SHOIN}(\mathbf{D})$. We assume a set of *elementary datatypes* and a set of *data values*. A *datatype* is either an elementary datatype or a set of data values (called *datatype oneOf*). A *datatype theory* $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a *datatype domain* $\Delta^{\mathbf{D}}$ and a mapping $\cdot^{\mathbf{D}}$ that assigns to each elementary datatype a subset of $\Delta^{\mathbf{D}}$ and to each data value an element of $\Delta^{\mathbf{D}}$. The mapping $\cdot^{\mathbf{D}}$ is extended to all datatypes by $\{v_1, \ldots\}^{\mathbf{D}} = \{v_1^{\mathbf{D}}, \ldots\}$. Let $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, and $\mathbf{I}$ be pairwise disjoint (nonempty) denumerable sets of *atomic concepts*, *abstract roles*, *datatype roles*, and *individuals*, respectively. We denote by $\mathbf{R}_A^-$ the set of inverses R^- of all $R \in \mathbf{R}_A$.

A *role* is an element of $\mathbf{R}_A \cup \mathbf{R}_A^- \cup \mathbf{R}_D$. *Concepts* are inductively defined as follows. Every $\phi \in \mathbf{A}$ is a concept, and if $o_1, \ldots, o_n \in \mathbf{I}$, then $\{o_1, \ldots, o_n\}$ is a concept (called *oneOf*). If ϕ, ϕ_1, and ϕ_2 are concepts and if $R \in \mathbf{R}_A \cup \mathbf{R}_A^-$, then also $(\phi_1 \sqcap \phi_2)$, $(\phi_1 \sqcup \phi_2)$, and $\neg\phi$ are concepts (called *conjunction*, *disjunction*, and *negation*, respectively), as well as $\exists R.\phi$, $\forall R.\phi$, $\geqslant nR$, and $\leqslant nR$ (called *exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geqslant 0$. If D is a datatype and $U \in \mathbf{R}_D$, then $\exists U.D$, $\forall U.D$, $\geqslant nU$, and $\leqslant nU$ are concepts (called *datatype exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geqslant 0$. We write $\top$ and $\bot$ to abbreviate the concepts $\phi \sqcup \neg\phi$ and $\phi \sqcap \neg\phi$, respectively, and we eliminate parentheses as usual.

An *axiom* has one of the following forms: (1) $\phi \sqsubseteq \psi$ (called *concept inclusion axiom*), where ϕ and ψ are concepts; (2) $R \sqsubseteq S$ (called *role inclusion axiom*), where either $R, S \in \mathbf{R}_A$ or $R, S \in \mathbf{R}_D$; (3) $\mathrm{Trans}(R)$ (called *transitivity axiom*), where $R \in \mathbf{R}_A$; (4) $\phi(a)$ (called *concept membership axiom*), where ϕ is a concept and $a \in \mathbf{I}$; (5) $R(a, b)$ (resp., $U(a, v)$) (called *role membership axiom*), where $R \in \mathbf{R}_A$ (resp., $U \in \mathbf{R}_D$) and $a, b \in \mathbf{I}$ (resp., $a \in \mathbf{I}$ and v is a data value); and (6) $a = b$ (resp., $a \neq b$) (*equality* (resp., *inequality*) *axiom*), where $a, b \in \mathbf{I}$. A *knowledge base* L is a finite set of axioms. For decidability, number restrictions in L are restricted to simple abstract roles [23].

The syntax of $\mathcal{SHIF}(\mathbf{D})$ is as the above syntax of $\mathcal{SHOIN}(\mathbf{D})$, but without the oneOf constructor and with the atleast and atmost constructors limited to 0 and 1.

Example 3.1. The subsequent description logic knowledge base L expresses that (1) textbooks are books, (2) personal computers and laptops are mutually exclusive electronic products, (3) books and electronic products are mutually exclusive products, (4) objects on offer are products, (5) every product has at least one related product, (6) only products are related to each other, (7) the relatedness between products is symmetric, (8) *tb_ai* and *tb_lp* are textbooks, (9) which are related to each other, (10) *pc_ibm* and *pc_hp* are personal computers, (11) which are related to each other, and (12) *ibm* and *hp* are providers for *pc_ibm* and *pc_hp*, respectively.

(1) $\mathit{textbook} \sqsubseteq \mathit{book}$; (2) $\mathit{pc} \sqcup \mathit{laptop} \sqsubseteq \mathit{electronics}$; $\mathit{pc} \sqsubseteq \neg \mathit{laptop}$;
(3) $\mathit{book} \sqcup \mathit{electronics} \sqsubseteq \mathit{product}$; $\mathit{book} \sqsubseteq \neg \mathit{electronics}$; (4) $\mathit{offer} \sqsubseteq \mathit{product}$;
(5) $\mathit{product} \sqsubseteq\, \geqslant 1\, \mathit{related}$; (6) $\geqslant 1\, \mathit{related} \sqcup \geqslant 1\, \mathit{related}^- \sqsubseteq \mathit{product}$;
(7) $\mathit{related} \sqsubseteq \mathit{related}^-$; $\mathit{related}^- \sqsubseteq \mathit{related}$;
(8) $\mathit{textbook}(\mathit{tb\_ai})$; $\mathit{textbook}(\mathit{tb\_lp})$; (9) $\mathit{related}(\mathit{tb\_ai}, \mathit{tb\_lp})$;
(10) $\mathit{pc}(\mathit{pc\_ibm})$; $\mathit{pc}(\mathit{pc\_hp})$; (11) $\mathit{related}(\mathit{pc\_ibm}, \mathit{pc\_hp})$;
(12) $\mathit{provides}(\mathit{ibm}, \mathit{pc\_ibm})$; $\mathit{provides}(\mathit{hp}, \mathit{pc\_hp})$.

Semantics. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ w.r.t. a datatype theory $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a nonempty (*abstract*) *domain* $\Delta^{\mathcal{I}}$ disjoint from $\Delta^{\mathbf{D}}$, and a mapping $\cdot^{\mathcal{I}}$ that assigns to each atomic concept $\phi \in \mathbf{A}$ a subset of $\Delta^{\mathcal{I}}$, to each individual $o \in \mathbf{I}$ an element of $\Delta^{\mathcal{I}}$, to each abstract role $R \in \mathbf{R}_A$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to each datatype role $U \in \mathbf{R}_D$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathbf{D}}$. We extend $\cdot^{\mathcal{I}}$ to all concepts and roles, and we define the *satisfaction* of an axiom F in an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, denoted $\mathcal{I} \models F$, as usual [19]. We say $\mathcal{I}$ *satisfies* the axiom F, or $\mathcal{I}$ is a *model* of F, iff $\mathcal{I} \models F$. We say $\mathcal{I}$ *satisfies* a knowledge base L, or $\mathcal{I}$ is a *model* of L, denoted $\mathcal{I} \models L$, iff $\mathcal{I} \models F$ for all $F \in L$. We say L is *satisfiable* (resp., *unsatisfiable*) iff L has a (resp., no) model. An axiom F is a *logical consequence* of L, denoted $L \models F$, iff each model of L satisfies F.

4 Disjunctive DL-Programs Under the Answer Set Semantics

In this section, we present a novel integration between disjunctive programs under the answer set semantics and description logics. The basic idea behind this integration is as follows. Suppose that we have a disjunctive program P. Under the answer set semantics, P is equivalent to its grounding $\mathit{ground}(P)$. Suppose now that some of the ground atoms in $\mathit{ground}(P)$ are additionally related to each other by a description logic knowledge base L. That is, some of the ground atoms in $\mathit{ground}(P)$ actually represent concept and role memberships relative to L. Thus, when processing $\mathit{ground}(P)$, we also have to consider L. However, we only want to do it to the extent that we actually need it for processing $\mathit{ground}(P)$. Hence, when taking a Herbrand interpretation $I \subseteq HB_{\Phi}$, we have to ensure that the ground atoms of I represent a valid constellation relative to L.

In other words, the main idea behind the semantics is to interpret P relative to Herbrand interpretations that also satisfy L, while L is interpreted relative to general interpretations over a first-order domain. Thus, we modularly combine the standard semantics of disjunctive programs and of description logics as in [11,12,10], which allows for building on the standard techniques and the results of both areas. However, our new approach here allows for a much tighter integration of L and P.

Syntax. We assume a function-free first-order vocabulary Φ with nonempty finite sets of constant and predicate symbols, as in Section 2. We use Φ_c to denote the set of all constant symbols in Φ. We also assume pairwise disjoint (nonempty) denumerable sets $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, and $\mathbf{I}$ of atomic concepts, abstract roles, datatype roles, and individuals, respectively, as in Section 3. We assume that Φ_c is a subset of $\mathbf{I}$. This assumption guarantees that every ground atom constructed from atomic concepts, abstract roles, datatype roles, and constants in Φ_c can be interpreted in the description logic component. We do not assume any other restriction on the vocabularies, that is, Φ and $\mathbf{A}$ (resp., $\mathbf{R}_A \cup \mathbf{R}_D$) may have unary (resp., binary) predicate symbols in common.

A *disjunctive description logic program* (or simply *disjunctive dl-program*) $KB = (L, P)$ consists of a description logic knowledge base L and a disjunctive program P. It is *positive* iff P is positive. It is a *normal dl-program* iff P is a normal program.

Example 4.1. A disjunctive dl-program $KB = (L, P)$ is given by the description logic knowledge base L and the disjunctive program P of Examples 2.1 and 3.1, respectively. Another disjunctive dl-program $KB' = (L', P')$ is obtained from KB by adding to L the axiom $\geqslant 1\, \mathit{similar} \sqcup \geqslant 1\, \mathit{similar}^- \sqsubseteq \mathit{product}$, which expresses that only products are similar. Observe that the predicate symbol *similar* in P' is also a role in L', and it freely occurs in both rule bodies and rule heads in P' (which is not possible in [11]).

Semantics. We now define the answer set semantics of disjunctive dl-programs via a generalization of the FLP-reduct of disjunctive programs (see Section 2).

In the sequel, let $KB = (L, P)$ be a disjunctive dl-program. A *ground instance* of a rule $r \in P$ is obtained from r by replacing every variable that occurs in r by a constant symbol from Φ_c. We denote by $ground(P)$ the set of all ground instances of rules in P. The *Herbrand base* relative to Φ, denoted HB_Φ, is the set of all ground atoms constructed with constant and predicate symbols from Φ. Observe that we now define the Herbrand base relative to Φ and not relative to P. This allows for reasoning about ground atoms from the description logic component that do not necessarily occur in P. Observe, however, that the extension from P to Φ is only a notational simplification, since we can always make constant and predicate symbols from Φ occur in P by "dummy" rules such as $\mathit{constant}(c) \leftarrow$ and $p(\boldsymbol{c}) \leftarrow p(\boldsymbol{c})$, respectively. We denote by DL_Φ the set of all ground atoms in HB_Φ that are constructed from atomic concepts in $\mathbf{A}$, abstract roles in $\mathbf{R}_A$, concrete roles in $\mathbf{R}_D$, and constant symbols in Φ_c.

An *interpretation* I is any subset of HB_Φ. We say I is a *model* of a description logic knowledge base L, denoted $I \models L$, iff $L \cup I \cup \{\neg a \mid a \in HB_\Phi - I\}$ is satisfiable. Note that the former defines the truth of description logic knowledge bases L in Herbrand interpretations $I \subseteq HB_\Phi$ rather than first-order interpretations $\mathcal{I}$. Note also that a negative concept membership $\neg C(a)$ can be encoded as the positive concept membership $(\neg C)(a)$. The following theorem shows that also negative role memberships $\neg R(b, c)$ can be reduced to positive concept memberships and concept inclusions.

Theorem 4.1. *Let L be a description logic knowledge base, and let $R(b, c)$ be a role membership axiom. Then, $L \cup \{\neg R(b, c)\}$ is satisfiable iff $L \cup \{B(b), C(c), \exists R.C \sqsubseteq \neg B\}$ is satisfiable, where B and C are two fresh atomic concepts.*

An interpretation $I \subseteq HB_\Phi$ is a *model* of a disjunctive dl-program $KB = (L, P)$, denoted $I \models KB$, iff $I \models L$ and $I \models P$. We say KB is *satisfiable* iff it has a model.

Given a disjunctive dl-program $KB = (L, P)$, the *FLP-reduct* of KB relative to an interpretation $I \subseteq HB_\Phi$, denoted KB^I, is the disjunctive dl-program (L, P^I), where P^I is the set of all $r \in ground(P)$ such that $I \models B(r)$. An interpretation $I \subseteq HB_\Phi$ is an *answer set* of KB iff I is a minimal model of KB^I. A disjunctive dl-program KB is *consistent* (resp., *inconsistent*) iff it has an (resp., no) answer set.

We finally define the notions of *cautious* (resp., *brave*) *reasoning* from disjunctive dl-programs under the answer set semantics as follows. A ground atom $a \in HB_{\Phi}$ is a *cautious* (resp., *brave*) *consequence* of a disjunctive dl-program KB under the answer set semantics iff every (resp., some) answer set of KB satisfies a.

5 Semantic Properties

In this section, we summarize some semantic properties (especially those relevant for the Semantic Web) of disjunctive dl-programs under the above answer set semantics.

Minimal Models. The following theorem shows that, like in the ordinary case (see Section 2), every answer set of a disjunctive dl-program KB is also a minimal model of KB, and the answer sets of a positive KB are given by the minimal models of KB.

Theorem 5.1. *Let $KB = (L, P)$ be a disjunctive dl-program. Then, (a) every answer set of KB is a minimal model of KB, and (b) if KB is positive, then the set of all answer sets of KB is given by the set of all minimal models of KB.*

Faithfulness. An important property of integrations of rules and ontologies is that they are a faithful [33,34] extension of both rules and ontologies.

The following theorem shows that the answer set semantics of disjunctive dl-programs faithfully extends its ordinary counterpart. That is, the answer set semantics of a disjunctive dl-program $KB = (L, P)$ with empty description logic knowledge base L coincides with the ordinary answer set semantics of its disjunctive program P.

Theorem 5.2. *Let $KB = (L, P)$ be a disjunctive dl-program such that $L = \emptyset$. Then, the set of all answer sets of KB coincides with the set of all ordinary answer sets of P.*

The next theorem shows that the answer set semantics of disjunctive dl-programs also faithfully extends the first-order semantics of description logic knowledge bases, which here means that a ground atom $a \in HB_{\Phi}$ is true in all answer sets of a positive disjunctive dl-program $KB = (L, P)$ iff a is true in all first-order models of $L \cup ground(P)$. The theorem holds also when a is a ground formula constructed from HB_{Φ} using $\wedge$ and $\vee$. Observe that the theorem does not hold for all first-order formulas a, but we actually also do not need this, *looking from the perspective of answer set programming*, since we actually cannot refer to all general first-order formulas in P.

Theorem 5.3. *Let $KB = (L, P)$ be a positive disjunctive dl-program, and let a be a ground atom from HB_{Φ}. Then, a is true in all answer sets of KB iff a is true in all first-order models of $L \cup ground(P)$.*

As an immediate corollary, we obtain that $a \in HB_{\Phi}$ is true in all answer sets of a disjunctive dl-program $KB = (L, \emptyset)$ iff a is true in all first-order models of L.

Corollary 5.1. *Let $KB = (L, P)$ be a disjunctive dl-program with $P = \emptyset$, and let $a \in HB_{\Phi}$. Then, a is true in all answer sets of KB iff a is true in all first-order models of L.*

Closed-World Assumption. It is often argued that the closed-world assumption is not very desirable in the open environment of the Semantic Web [20]. The notion of cautious reasoning from disjunctive dl-programs under the answer set semantics also has some closed-world property. However, as also shown by Theorem 5.3, this closed-world property is actually limited to the explicit use of default negations in rule bodies, and thus we can actually control very easily its use in disjunctive dl-programs.

Unique Name Assumption. Another aspect that may not be very desirable in the Semantic Web [20] is the *unique name assumption* (which says that any two distinct constant symbols in Φ_c represent two distinct domain objects). It turns out that we actually do not have to make this assumption, since the description logic knowledge base of a disjunctive dl-program may very well contain or imply equalities between individuals.

This result is included in the following theorem, which shows an alternative characterization of the satisfaction of L in $I \subseteq HB_\Phi$: Rather than being enlarged by a set of axioms of exponential size, L is enlarged by a set of axioms of polynomial size. This characterization essentially shows that the satisfaction of L in I corresponds to checking that (i) the ground atoms in $I \cap DL_\Phi$ satisfy L, and (ii) the ground atoms in $I \cap (HB_\Phi - DL_\Phi)$ do not violate any equality axioms that follow from L. Here, an equivalence relation $\sim$ on Φ_c is *admissible* with an interpretation $I \subseteq HB_\Phi$ iff $p(c_1, \ldots, c_n) \in I \Leftrightarrow p(c'_1, \ldots, c'_n) \in I$ for all n-ary predicate symbols p, where $n > 0$, and constant symbols $c_1, \ldots, c_n, c'_1, \ldots, c'_n \in \Phi_c$ such that $c_i \sim c'_i$ for all $i \in \{1, \ldots, n\}$.

Theorem 5.4. *Let L be a description logic knowledge base, and let $I \subseteq HB_\Phi$. Then, $L \cup I \cup \{\neg b \mid b \in HB_\Phi - I\}$ is satisfiable iff $L \cup (I \cap DL_\Phi) \cup \{\neg b \mid b \in DL_\Phi - I\} \cup \{c \neq c' \mid c \not\sim c'\}$ is satisfiable for some equivalence relation $\sim$ on Φ_c admissible with I.*

Conjunctive Queries. It is often argued that the processing of conjunctive queries is important for the Semantic Web [37]. As for this issue, observe that (Boolean unions of) conjunctive queries in our approach can be reduced to atomic queries. A *Boolean union of conjunctive queries* Q is of the form $\exists \boldsymbol{x}(\gamma_1(\boldsymbol{x}) \vee \cdots \vee \gamma_n(\boldsymbol{x}))$, where $\boldsymbol{x}$ is a tuple of variables, $n \geqslant 1$, and each $\gamma_i(\boldsymbol{x})$ is a conjunction of atoms constructed from predicate and constant symbols in Φ and variables in $\boldsymbol{x}$. We call Q a *conjunctive query* when $n = 1$. If we assume that $\boldsymbol{x}$ ranges over all constant symbols in Φ_c (which is sufficient for our needs, *looking from the perspective of answer set programming*, since in P we can refer only through Φ_c to elements of a first-order domain), then Q can be expressed by adding the rules $q(\boldsymbol{x}) \leftarrow \gamma_i(\boldsymbol{x})$ with $i \in \{1, \ldots, n\}$ to P and thereafter computing the set of all entailed ground instances of $q(\boldsymbol{x})$ relative to Φ_c (see also Section 6).

6 Algorithms and Complexity

In this section, we describe algorithms for deciding whether a disjunctive dl-program has an answer set, and for deciding brave and cautious consequences from disjunctive dl-programs under the answer set semantics. Furthermore, we also draw a precise picture of the complexity of all these decision problems.

Algorithm consistency

Input: vocabulary Φ and disjunctive dl-program $KB = (L, P)$.
Output: *Yes*, if KB has an answer set; *No*, otherwise.

1. **if** there exists $I \subseteq HB_{\Phi}$ such that I is a minimal model of $KB^I = (L, P^I)$
2. **then return** *Yes*;
3. **else return** *No*.

Fig. 1. Algorithm consistency

Algorithms. The problem of deciding whether a disjunctive dl-program $KB = (L, P)$ has an answer set can be solved by a simple guess-and-check algorithm, which guesses a subset I of the finite Herbrand base HB_{Φ}, computes the FLP-reduct $KB^I = (L, P^I)$, and then checks that I is in fact a minimal model of KB^I (see Fig. 1).

The problem of deciding brave and cautious consequences can be reduced to deciding answer set existence (like in the ordinary case), since a ground atom $a \in HB_{\Phi}$ is true in some (resp., every) answer set of a disjunctive dl-program $KB = (L, P)$ iff $(L, P \cup \{\leftarrow \mathit{not}\, a\})$ (resp., $(L, P \cup \{\leftarrow a\})$) has an (resp., no) answer set.

Complexity. We now show that the problems of deciding consistency and brave / cautious consequences have the same complexity in disjunctive dl-programs under the answer set semantics as in ordinary disjunctive programs under the answer set semantics.

The following theorem shows that deciding the consistency of disjunctive dl-programs is complete for $\mathrm{NEXP}^{\mathrm{NP}}$ (combined complexity). The lower bound follows from the $\mathrm{NEXP}^{\mathrm{NP}}$-hardness of deciding the consistency of ordinary disjunctive programs [6]. The upper bound follows from the result that deciding knowledge base satisfiability in $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$) is complete for EXP (resp., NEXP) [40,19].

Theorem 6.1. *Given Φ and a disjunctive dl-program $KB = (L, P)$ with L in $\mathcal{SHIF}(\mathbf{D})$ or $\mathcal{SHOIN}(\mathbf{D})$, deciding whether KB has an answer set is complete for* $\mathrm{NEXP}^{\mathrm{NP}}$.

The next theorem shows that deciding cautious (resp., brave) consequences from disjunctive dl-programs is complete for $\text{co-}\mathrm{NEXP}^{\mathrm{NP}}$ (resp., $\mathrm{NEXP}^{\mathrm{NP}}$) in the combined complexity. This result follows from Theorem 6.1, since the two problems of consistency checking and cautious (resp., brave) reasoning can be reduced to each other.

Theorem 6.2. *Given Φ, a disjunctive dl-program $KB = (L, P)$ with L in $\mathcal{SHIF}(\mathbf{D})$ or $\mathcal{SHOIN}(\mathbf{D})$, and a ground atom $a \in HB_{\Phi}$, deciding whether a holds in every (resp., some) answer set of KB is complete for* $\text{co-}\mathrm{NEXP}^{\mathrm{NP}}$ *(resp.,* $\mathrm{NEXP}^{\mathrm{NP}}$*).*

7 Tractability Results

In this section, we describe a special class of disjunctive dl-programs for which the problems of deciding consistency and of query processing have both a polynomial data complexity. These programs are normal, stratified, and defined relative to *DL-Lite* [5], which allows for deciding knowledge base satisfiability in polynomial time.

We first recall *DL-Lite*. Let $\mathbf{A}$, $\mathbf{R}_A$, and $\mathbf{I}$ be pairwise disjoint sets of atomic concepts, abstract roles, and individuals, respectively. A *basic concept in DL-Lite* is either an atomic concept from $\mathbf{A}$ or an exists restriction on roles $\exists R.\top$ (abbreviated as $\exists R$), where $R \in \mathbf{R}_A \cup \mathbf{R}_A^-$. A *literal in DL-Lite* is either a basic concept b or the negation of a basic concept $\neg b$. *Concepts in DL-Lite* are defined by induction as follows. Every basic concept in *DL-Lite* is a concept in *DL-Lite*. If b is a basic concept in *DL-Lite*, and ϕ_1 and ϕ_2 are concepts in *DL-Lite*, then $\neg b$ and $\phi_1 \sqcap \phi_2$ are also concepts in *DL-Lite*. An *axiom in DL-Lite* is either (1) a concept inclusion axiom $b \sqsubseteq \psi$, where b is a basic concept in *DL-Lite* and ϕ is a concept in *DL-Lite*, or (2) a *functionality axiom* $(\mathsf{funct}\ R)$, where $R \in \mathbf{R}_A \cup \mathbf{R}_A^-$, or (3) a concept membership axiom $b(a)$, where b is a basic concept in *DL-Lite* and $a \in \mathbf{I}$, or (4) a role membership axiom $R(a, c)$, where $R \in \mathbf{R}_A$ and $a, c \in \mathbf{I}$. A *knowledge base in DL-Lite* L is a finite set of axioms in *DL-Lite*.

Every knowledge base in *DL-Lite* L can be transformed into an equivalent one in *DL-Lite* $trans(L)$ in which every concept inclusion axiom is of form $b \sqsubseteq \ell$, where b (resp., ℓ) is a basic concept (resp., literal) in *DL-Lite* [5]. We then define $trans(P) = P \cup \{b'(X) \leftarrow b(X) \mid b \sqsubseteq b' \in trans(L),\ b'$ is a basic concept$\} \cup \{\exists R(X) \leftarrow R(X, Y) \mid R \in \mathbf{R}_A \cap \Phi\} \cup \{\exists R^-(Y) \leftarrow R(X, Y) \mid R \in \mathbf{R}_A \cap \Phi\}$. Intuitively, we make explicit all the rule-based relationships between the predicates in P that are implicitly encoded in L. We define stratified normal dl-programs as follows. A normal dl-program $KB = (L, P)$ is *stratified* iff (i) L is defined in *DL-Lite* and (ii) $trans(P)$ is (locally) stratified.

It can be shown that stratified normal dl-programs $KB = (L, P)$ have either no or a unique answer set, which can be computed by a finite sequence of fixpoint iterations (relative to $trans(P)$), as usual. This implies immediately that for such programs consistency checking and query processing have both a polynomial data complexity.

Theorem 7.1. *Given Φ and a stratified normal dl-program KB, (a) deciding whether KB has an answer set, and (b) deciding whether a given ground atom $a \in HB_\Phi$ is true in the answer set of KB (if it exists) have both a polynomial data complexity.*

8 Related Work

There is a large body of related works on combining rules and ontologies, which can essentially be divided into the following three lines of research: (a) loose integration of rules and ontologies, (b) tight integration of rules and ontologies, and (c) reductions from description logics to logic programming formalisms. In this section, we discuss only the works that are most closely related to the framework of this paper.

Representatives of the loose integration of rules and ontologies are in particular the dl-programs in [11,12], their extension to HEX-programs [9,10], to probabilistic dl-programs [28,29], and to fuzzy dl-programs [30]. The combination of defeasible reasoning with description logics in [3], the calls to description logic reasoners in TRIPLE [38], and the hybrid MKNF knowledge bases in [33,34] are also close in spirit. More concretely, compared to the present paper, the dl-programs $KB = (L, P)$ in [11] also consist of a description logic knowledge base L and a normal program P. However, P may also contain classical negations, and rather than using concepts and roles from L as predicates in P, rule bodies in P may only contain queries to L, which may also contain facts as additional input to L. Like in this paper, P is interpreted relative to

Herbrand interpretations under the answer set semantics, while L is interpreted relative to first-order interpretations under the classical model-theoretic semantics. However, differently from the concepts and roles in P here, the queries in P in [11] are evaluated independently from each other. HEX-programs [9,10] extend the approach to dl-programs in [11] by multiple sources of external knowledge, with possibly different semantics, while probabilistic dl-programs [28,29] and fuzzy dl-programs [30] are extensions by probabilistic uncertainty and fuzzy vagueness, respectively. Closely related to the dl-programs in [11] are also the hybrid MKNF knowledge bases in [33,34]. They essentially allow for querying a description logic knowledge base L via the operators $\mathbf{K}$ and $\mathbf{not}$, which can be used more flexibly than the queries in [11] (the operators can also occur in rule heads, while the queries are restricted to rule bodies), but which do not allow for passing facts to L in the form of query arguments. Note that closely related to the hybrid MKNF knowledge bases in [33,34] is also the embedding of non-ground logic programs into autoepistemic logic in [7]. The following example shows that our novel dl-programs here generally do not have the same meaning as the dl-programs in [11] (note that a similar example can be constructed for the approach in [33,34]).

Example 8.1. The normal dl-program $KB = (L, P)$, where

$$\begin{aligned} L &= \{\mathit{person}(a),\ \mathit{person} \sqsubseteq \mathit{male} \sqcup \mathit{female}\} \text{ and} \\ P &= \{\mathit{client}(X) \leftarrow \mathit{male}(X),\ \mathit{client}(X) \leftarrow \mathit{female}(X)\} \end{aligned}$$

implies $\mathit{client}(a)$, while the normal dl-program $KB' = (L', P')$ as in [11]

$$\begin{aligned} L' &= \{\mathit{person}(a),\ \mathit{person} \sqsubseteq \mathit{male} \sqcup \mathit{female}\} \text{ and} \\ P' &= \{\mathit{client}(X) \leftarrow \mathit{DL}[\mathit{male}](X),\ \mathit{client}(X) \leftarrow \mathit{DL}[\mathit{female}](X)\} \end{aligned}$$

does *not* imply $\mathit{client}(a)$, since the two queries are evaluated independently from each other, and neither $\mathit{male}(a)$ nor $\mathit{female}(a)$ follows from L'. To obtain the conclusion $\mathit{client}(a)$ in [11], one has to directly use the rule $\mathit{client}(X) \leftarrow \mathit{DL}[\mathit{male} \sqcup \mathit{female}](X)$.

Some representatives of tight integrations of rules and ontologies are in particular the works due to Donini et al. [8], Levy and Rousset [27], Grosof et al. [16], Motik et al. [35], Heymans et al. [17], and Rosati [36,37]. SWRL [21] and WRL [2] also belong to this category. Closest in spirit to this paper among the above works is perhaps Rosati's approach [36,37]. Like here, Rosati's hybrid knowledge bases also consist of a description logic knowledge base L and a disjunctive program (with default negations) P, where concepts and roles in L may act as predicate symbols in P. However, differently from this paper, Rosati partitions the predicates of L and P into description logic predicates and logic program predicates, where the former are interpreted under the classical model-theoretic semantics, while the latter are interpreted under the answer set semantics (and thus in particular default negations of concepts and roles are not allowed in P). Furthermore, differently from this paper, he also assumes a syntactic restriction on rules (called *weak safeness*) to gain decidability, and he assumes the standard names assumption, which includes the unique name assumption.

Finally, the works reducing description logics to logic programming are less closely related to the framework of this paper. Some representatives are in particular the works by Alsaç and Baral [1], Swift [39], Heymans and Vermeir [18], and Motik et al. [24].

9 Summary and Outlook

We have presented a novel combination of disjunctive programs under the answer set semantics with description logics for the Semantic Web. The combination is based on a well-balanced interface between disjunctive programs and description logics, which guarantees the decidability of the resulting formalism without assuming any syntactic restrictions on the resulting language (such as syntactic safety conditions and/or syntactic partitionings of the vocabulary). We have shown that the new formalism has very nice semantic properties. In particular, it faithfully extends both disjunctive programs and description logics. We have also provided algorithms and precise complexity results for the new formalism, as well as a special case of polynomial data complexity.

The presented mechanism of integrating rules and ontologies is of general importance, since it can actually also be used for the decidable integration of other reasoning techniques (such as reasoning about defaults, probabilistic uncertainty, and fuzzy vagueness) with description logics, since it applies to all reasoning techniques that are based on interpretations over finite Herbrand bases (or also finite sets of propositional symbols). It thus paves the way for decidable reasoning formalisms on top of description logics for the Semantic Web. Note that a companion paper [32] explores the use of this novel integration mechanism in fuzzy description logic programs.

An interesting topic of future research is to develop more sophisticated algorithms for reasoning from the new disjunctive dl-programs, and to implement the approach. Another interesting issue is to extend disjunctive dl-programs by classical negation.

Acknowledgments. This work has been supported by a Heisenberg Professorship of the German Research Foundation (DFG). I thank the reviewers for their constructive comments, which helped to improve this work.

References

1. G. Alsaç and C. Baral. Reasoning in description logics using declarative logic programming. Report, Department of Computer Science and Engineering, Arizona State University, 2001.
2. J. Angele, H. Boley, J. de Bruijn, D. Fensel, P. Hitzler, M. Kifer, R. Krummenacher, H. Lausen, A. Polleres, and R. Studer. Web Rule Language (WRL), Sept. 2005. W3C Member Submission. http://www.w3.org/Submission/WRL/.
3. G. Antoniou. Nonmonotonic rule systems on top of ontology layers. In *Proc. ISWC-2002*, pp. 394–398, 2002.
4. T. Berners-Lee. *Weaving the Web*. Harper, San Francisco, CA, 1999.
5. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. *DL-Lite*: Tractable description logics for ontologies. In *Proc. AAAI-2005*, pp. 602–607, 2005.
6. E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Comput. Surv.*, 33(3):374–425, 2001.
7. J. de Bruijn, T. Eiter, A. Polleres, and H. Tompits. Embedding non-ground logic programs into autoepistemic logic for knowledge-base combination. In *Proc. IJCAI-2007*, pp. 304–309, 2007.
8. F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. $\mathcal{AL}$-log: Integrating datalog and description logics. *J. Intell. Inf. Syst.*, 10(3):227–252, 1998.

9. T. Eiter, G. Ianni, R. Schindlauer, H. Tompits. A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In *Proc. IJCAI-2005*, pp. 90–96.
10. T. Eiter, G. Ianni, R. Schindlauer, H. Tompits. Effective integration of declarative rules with external evaluations for semantic web reasoning. In *Proc. ESWC-2006*, pp. 273–287, 2006.
11. T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the Semantic Web. In *Proc. KR-2004*, pp. 141–151, 2004.
12. T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Well-founded semantics for description logic programs in the Semantic Web. In *Proc. RuleML-2004*, pp. 81–97, 2004.
13. W. Faber, N. Leone, and G. Pfeifer. Recursive aggregates in disjunctive logic programs: Semantics and complexity. In *Proc. JELIA-2004*, pp. 200–212, 2004.
14. D. Fensel, W. Wahlster, H. Lieberman, and J. Hendler, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2002.
15. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Comput.*, 9(3/4):365–386, 1991.
16. B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: Combining logic programs with description logics. In *Proc. WWW-2003*, pp. 48–57, 2003.
17. S. Heymans, D. V. Nieuwenborgh, D. Vermeir. Nonmonotonic ontological and rule-based reasoning with extended conceptual logic programs. In *Proc. ESWC-05*, pp. 392–407, 2005.
18. S. Heymans and D. Vermeir. Integrating semantic web reasoning and answer set programming. In *Proc. ASP-2003*, pp. 194–208, 2003.
19. I. Horrocks and P. F. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. In *Proc. ISWC-2003*, pp. 17–29, 2003.
20. I. Horrocks and P. F. Patel-Schneider. Position paper: A comparison of two modelling paradigms in the Semantic Web. In *Proc. WWW-2006*, pp. 3–12, 2006.
21. I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web rule language combining OWL and RuleML, May 2004. W3C Member Submission. Available at http://www.w3.org/Submission/SWRL/.
22. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. Web Sem.*, 1(1):7–26, 2003.
23. I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proc. LPAR-1999*, pp. 161–180, 1999.
24. U. Hustadt, B. Motik, and U. Sattler. Reducing SHIQ-description logic to disjunctive datalog programs. In *Proc. KR-2004*, pp. 152–162, 2004.
25. L. Iocchi, T. Lukasiewicz, D. Nardi, and R. Rosati. Reasoning about actions with sensing under qualitative and probabilistic uncertainty. In *Proc. ECAI-2004*, pp. 818–822, 2004.
26. N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. The DLV system for knowledge representation and reasoning. *ACM TOCL*, 7(3):499–562, 2006.
27. A. Y. Levy and M.-C. Rousset. Combining Horn rules and description logics in CARIN. *Artif. Intell.*, 104(1–2):165–209, 1998.
28. T. Lukasiewicz. Stratified probabilistic description logic programs. In *Proc. URSW-2005*, pp. 87–97, 2005.
29. T. Lukasiewicz. Probabilistic description logic programs. In *Proc. ECSQARU-2005*, pp. 737–749, 2005. Extended version in *Int. J. Approx. Reasoning*, in press.
30. T. Lukasiewicz. Fuzzy description logic programs under the answer set semantics for the Semantic Web. In *Proc. RuleML-2006*, pp. 89–96, 2006.
31. T. Lukasiewicz. A novel combination of answer set programming with description logics for the Semantic Web. Report 1843-06-08, Institut für Informationssysteme, TU Wien, 2006.
32. T. Lukasiewicz and U. Straccia. Tightly integrated fuzzy description logic programs under the answer semantics for the Semantic Web. In *Proc. RR-2007*, 2007. To appear.
33. B. Motik, I. Horrocks, R. Rosati, and U. Sattler. Can OWL and logic programming live together happily ever after? In *Proc. ISWC-2006*, pp. 501–514, 2006.

34. B. Motik and R. Rosati. A faithful integration of description logics with logic programming. In *Proc. IJCAI-2007*, pp. 477–482, 2007.
35. B. Motik, U. Sattler, and R. Studer. Query answering for OWL-DL with rules. *J. Web Sem.*, 3(1):41–60, 2005.
36. R. Rosati. On the decidability and complexity of integrating ontologies and rules. *J. Web Sem.*, 3(1):61–73, 2005.
37. R. Rosati. $\mathcal{DL}+log$: Tight integration of description logics and disjunctive datalog. In *Proc. KR-2006*, pp. 68–78, 2006.
38. M. Sintek and S. Decker. TRIPLE - A query, inference, and transformation language for the Semantic Web. In *Proc. ISWC-2002*, pp. 364–378, 2002.
39. T. Swift. Deduction in ontologies via ASP. In *Proc. LPNMR-2004*, pp. 275–288, 2004.
40. S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, Germany, 2001.
41. W3C. OWL web ontology language overview, 2004. W3C Recommendation (10 Feb. 2004). Available at http://www.w3.org/TR/2004/REC-owl-features-20040210/.

Appendix: Selected Proofs

Proof of Theorem 5.2. Observe first that $I \subseteq HB_\Phi$ is a model of $KB^I = (L, P^I)$ iff (i) $I \models L$ and (ii) $I \models r$ for every $r \in P^I$. Since $L = \emptyset$, this is equivalent to $I \models r$ for every $r \in P^I$. Thus, $I \subseteq HB_\Phi$ is a minimal model of KB^I iff I is a minimal model of P^I. That is, $I \subseteq HB_\Phi$ is an answer set of KB iff I is an ordinary answer set of P. □

Proof of Theorem 5.3. Observe first that, by Theorem 5.1, since P is positive, the set of all answer sets of KB is given by the set of all minimal models $I \subseteq HB_\Phi$ of KB. Observe then that $a \in HB_\Phi$ is true in all minimal models $I \subseteq HB_\Phi$ of KB iff a is true in all models $I \subseteq HB_\Phi$ of KB. It thus remains to show that a is true in all models $I \subseteq HB_\Phi$ of KB iff a is true in all first-order models of $L \cup ground(P)$:

($\Rightarrow$) Suppose that a is true in all models $I \subseteq HB_\Phi$ of KB. Let $\mathcal{I}$ be any first-order model of $L \cup ground(P)$. Let $I \subseteq HB_\Phi$ be defined by $b \in I$ iff $\mathcal{I} \models b$. Then, $\mathcal{I}$ is a model of $L^\star = L \cup I \cup \{\neg b \mid b \in HB_\Phi - I\}$, and thus $L^\star$ is satisfiable. Hence, I is a model of L. Since $\mathcal{I}$ is a model of $ground(P)$, also I is a model of $ground(P)$. In summary, this shows that I is a model of KB. Hence, a is true in I, and thus a is true in $\mathcal{I}$. Overall, this shows that a is true in all first-order models of $L \cup ground(P)$.

($\Leftarrow$) Suppose that a is true in all first-order models of $L \cup ground(P)$. Let $I \subseteq HB_\Phi$ be any model of KB. Then, $L^\star = L \cup I \cup \{\neg b \mid b \in HB_\Phi - I\}$ is satisfiable. Let $\mathcal{I}$ be a first-order model of $L^\star$. Then, $\mathcal{I}$ is in particular a model of L. Furthermore, since I is a model of $ground(P)$, also $\mathcal{I}$ is a model of $ground(P)$. In summary, $\mathcal{I}$ is a model of $L \cup ground(P)$. It thus follows that a is true in $\mathcal{I}$, and thus a is also true in I. Overall, this shows that a is true in all models $I \subseteq HB_\Phi$ of KB. □