

---

# Ontologies and Rules

Pascal Hitzler<sup>1</sup> and Bijan Parsia<sup>2</sup>

<sup>1</sup> AIFB, Universität Karlsruhe (TH), Germany, [hitzler@aifb.uni-karlsruhe.de](mailto:hitzler@aifb.uni-karlsruhe.de)

<sup>2</sup> University of Manchester, UK, [bparsia@cs.man.ac.uk](mailto:bparsia@cs.man.ac.uk)

## 1 Introduction

The Web Ontology Language OWL, as introduced in Chapter 4, is the language recommended by the World Wide Web consortium (W3C) for expressing ontologies for the Semantic Web. OWL is based on Description Logics, see Chapter 1, and as such is based on first-order predicate logic as underlying knowledge representation and reasoning paradigm.

Throughout the advent of the Semantic Web, however, F-Logic as an alternative approach for expressing ontologies has been based on rules, more precisely on the logic programming paradigm, see Chapter 2. Due to this, and also due to the importance of rule-based systems in industrial practice, it is natural to ask how OWL and rule-based knowledge-representation and reasoning can be combined.

Achieving a conceptually clear integration of OWL and rules, however, is not a trivial task. The reason for the difficulties lies in the fact that OWL – based on first-order predicate logic – adheres to the open-world assumption, while rules generally follow the closed-world assumption. Consequently, the semantics of OWL and rules differs considerably, and achieving a meaningful, intuitive, and formally clear combined semantics is not straightforward.

Research on the topic of ontologies and rules has thus spawned into several different directions. In this chapter, we provide an overview of the state of the art, by discussing briefly some of the approaches which we consider to be most promising or interesting. Since we want to be brief and to the point, our selection is naturally subjective, but we made an effort to include references to many of the recent publications on the topic.

In Section 2, we present the Semantic Web Rules Language SWRL, a rule *extension* of OWL DL which adheres to the open-world paradigm, and is thus entirely in the spirit of the OWL DL language. We also introduce the decidable fragment of SWRL known as DL-Safe rules. SWRL adds to the

expressive power of OWL by allowing the modelling of certain axioms which lie outside the capability of OWL DL.

In Section 3, we change perspective and consider rule *fragments* of OWL DL. In particular, we present its naive Horn fragment DLP – Description Logic Programs – and a more sophisticated fragment called Horn-*SHIQ*, which encompasses DLP.

In Section 4, we briefly address *hybrid* approaches, i.e. theories and systems which combine OWL with some existing rules language, and thus combine the open- and the closed-world assumption.

We conclude in Section 5.

Throughout the chapter, we will employ the syntax for Description Logics as introduced in Chapter 1.

### Acknowledgements

The first named author acknowledges support by the Deutsche Forschungsgemeinschaft under the ReaSem project.

## 2 SWRL and DL-Safe Rules

Attempts to combine some sort of rules with a description logic go back at least as far as the Classic system [2]. Starting in the late 1990s, there were a number of attempts to combine Datalog (as the premier deductive database language) with description logics; notable examples include  $\mathcal{AL}$ -log [6] and CARIN [22]. While partially motivated by the desire to increase the expressive power of both components, these attempts (esp.  $\mathcal{AL}$ -log) at hybrid systems were strongly constrained by the desire to retain both the modeling and the computational properties of the respective components, and even, perhaps, the implementation techniques or actual implementations. (For example, the proof procedure for  $\mathcal{AL}$ -log described in [6] calls an independently developed description logic reasoner as a oracle.)

In general, when considering combination formalisms, the upper bound is the unrestricted union of the two systems. In essence, the Semantic Web Rule Language (SWRL)<sup>3</sup> [13] is the unrestricted union of OWL DL (i.e., roughly the description logic  $\mathcal{SHOIN}(D)$ ) and (binary) function-free Horn logic. The result is a very expressive formalism which is, unsurprisingly, undecidable. Also, it is comparatively not well understood: there are no native reasoners for it, relationships to other formalisms are not precisely mapped out, and there is almost no experience in using SWRL for ontology modeling. However, it does serve as a unifying overarching formalism for various rule and rule like extension to OWL DL. It is, for example, a superset of (binary)  $\mathcal{AL}$ -log and of CARIN.

<sup>3</sup> <http://www.w3.org/Submission/SWRL/>

Decidability can be regained with the imposition of a *safety* condition on SWRL rules. Essentially, the possible values of (explicit) variables in SWRL rules are restricted to named individuals only which confines the direct effects of such rules to the ABox. This safety condition is known as “DL-Safety” and such SWRL rules are generally called “DL-Safe SWRL rules” or “DL-Safe rules.” Not only are DL-Safe rules (in combination with OWL-DL) decidable, but reasonable implementations are emerging.

SWRL was first proposed under the name OWL Rules Language [13] before gaining built-in atoms and a more RuleML<sup>4</sup> flavored syntax. In that form, it was published as a W3C member submission. The decidability of SWRL rules with the DL-Safety condition was established in [32, 33], and further elaborated in [29], including a discussion of a robust implementation of (most of) OWL plus DL-safe rules, KAON2.

## 2.1 Definition of SWRL

SWRL contains OWL DL as a proper part, that is, all OWL DL axioms are SWRL axioms. Additionally, a SWRL knowledge base may contain a set of *rules*, which consist of an antecedent (body) and a consequent (head) which themselves are sets of SWRL atoms.

A SWRL atom may be of the following<sup>5</sup> forms:

- Unary atoms:
  - $C(arg1)$  where C is an arbitrary OWL DL class expression
  - $D(dataArg1)$  where D is a datatype URI or an enumerated value range
- Binary atoms:
  - $P(arg1, arg2)$  where P is an object property
  - $Q(arg1, dataArg1)$  where Q is a datatype property
  - $arg1 = arg1$  equality, or “sameAs”
  - $arg2 \neq arg2$  inequality, or “differentFrom”

Where arguments are of the form:

$arg1 \mid arg2$  these are either individuals denoting URIs or individual ranging variables  
 $dataArg1$  these are data literals or data value ranging variables

A SWRL rule is of the form:

- $Atom_1 \wedge \dots \wedge Atom_n \rightarrow Atom_{n+1} \wedge \dots \wedge Atom_m$

where atoms 1 through  $n$  form the antecedent (i.e., body) and atoms  $n + 1$  through  $m$  form the consequent (i.e., the head).

<sup>4</sup> <http://www.ruleml.org/>

<sup>5</sup> In addition to the forms directly below, SWRL also allows for “built-in” atoms which are discussed in section 2.3.

The semantics of SWRL are traditionally given via an extension of the “direct” model theory for OWL DL<sup>6</sup>. However, for current purposes it is a little more perspicuous to present the semantics as an extension of the standard translation<sup>7</sup> of description logics to first order logic. To avoid various tedious details of the translation of  $\mathcal{SHOIN}(D)$  which are irrelevant to understanding SWRL, we give only the translation of the simpler description logic  $\mathcal{SH}$ , that is, omitting nominals, inverse roles, number restrictions, and datatypes (but retaining (in)equality). We therefore also eliminate SWRL atoms with data valued arguments.

In the table below,  $\mathcal{X}$  and  $\mathcal{Y}$  are meta-linguistic variables that range over constants (i.e., names of individuals) or object variables (i.e.,  $x$ ,  $y$ , and  $z$ ).  $A$  represents an arbitrary atomic concept,  $C$  and  $D$  represent arbitrary class expressions,  $a$  and  $b$  represent arbitrary constants, and  $P$  represents an arbitrary role:

- **Atomics**

Term	Translation
$\pi(x)$ or $\pi(y)$	$y$ or $x$ (resp.)
$\pi(A, \mathcal{X})$	$A(\mathcal{X})$
$\pi(P, \mathcal{X}, \mathcal{Y})$	$P(\mathcal{X}, \mathcal{Y})$
- **Concept expressions**

Concept	Translation
$\pi(A, \mathcal{X})$	$A(\mathcal{X})$
$\pi(\neg C, \mathcal{X})$	$\neg\pi(C, \mathcal{X})$
$\pi(C \sqcap D, \mathcal{X})$	$\pi(C, \mathcal{X}) \wedge \pi(D, \mathcal{X})$
$\pi(C \sqcup D, \mathcal{X})$	$\pi(C, \mathcal{X}) \vee \pi(D, \mathcal{X})$
$\pi(\exists P.C, \mathcal{X})$	$\exists\pi(\mathcal{X})(\pi(P, \mathcal{X}, \pi(\mathcal{X})) \wedge \pi(C, \pi(\mathcal{X})))$
$\pi(\forall P.C, \mathcal{X})$	$\forall\pi(\mathcal{X})(\pi(P, \mathcal{X}, \pi(\mathcal{X})) \rightarrow \pi(C, \pi(\mathcal{X})))$
- **Axioms<sup>8</sup>**

Axiom	Translation
$\Pi(C \sqsubseteq D)$	$\forall x(\pi(C, x) \rightarrow \pi(D, x))$
$\Pi(P \sqsubseteq Q)$	$\forall x, y(\pi(P, x, y) \rightarrow \pi(Q, x, y))$
$\Pi(Trans(P))$	$\forall x, y, z(\pi(P, x, y) \wedge \pi(P, y, z) \rightarrow \pi(P, x, z))$
$\Pi(a : C)$	$\pi(C, a)$
$\Pi(\langle a, b \rangle : P)$	$\pi(P, a, b)$
$\Pi(a = b)$	$a = b$
$\Pi(a \neq b)$	$a \neq b$

It is clear that, except for transitivity axioms,  $\mathcal{SH}$  can be encoded into first order logic using only two distinct variables (i.e.,  $x$  and  $y$ ) and transitivity

<sup>6</sup> As in the SWRL submission.

<sup>7</sup> For a basic discussion of the standard translation in a modal logic context, see chapter 2 of [1]. For variants of the standard translation for  $\mathcal{SHOIN}(D)$  and other description logics see [33] and [16].

<sup>8</sup> Class and property equivalence axioms can be defined as a pair of inclusion axioms.

only needs one additional variable (i.e.,  $z$ ). To extend this translation to a SWRL extension of  $\mathcal{SH}$ , we need an arbitrary supply of variables, that is, at least enough for each distinct variable in the rule with the largest number of distinct variables in a given rule set. To help distinguish translations of  $\mathcal{SH}$  axioms from translations of SWRL rules, we shall use capital letters from  $\{X, Y, Z\}$  with subscripts when necessary.

Let  $\Psi = \psi_1 \wedge \dots \wedge \psi_n \rightarrow \psi_{n+1} \wedge \dots \wedge \psi_m$  be a SWRL rule, such that  $Var(\Psi) = V_1 \dots V_k$  are all the SWRL variables in the atoms  $\psi_1 \dots \psi_m$ . Then we can extend the translation function,  $\pi$ , as follows (note that the meta-variables  $\mathcal{X}$  and  $\mathcal{Y}$  now also range over SWRL variables):

- **Atomics**

Term	Translation
$\pi(x)$ or $\pi(y)$ or $\pi(\mathcal{V})$	$y$ or $x$ or $\mathcal{V}$ (resp.)
$\pi(A, \mathcal{X})$	$A(\mathcal{X})$
$\pi(P, \mathcal{X}, \mathcal{Y})$	$P(\mathcal{X}, \mathcal{Y})$
- **Axioms**

Axiom	Translation
$\pi(\Psi)$	$\forall V_1 \dots V_k (\pi(\psi_1) \wedge \dots \wedge \pi(\psi_n) \rightarrow \pi(\psi_{n+1}) \wedge \dots \wedge \pi(\psi_m))$ Where a $\psi$ is of the form $A(\mathcal{X})$ or $P(\mathcal{X}, \mathcal{Y})$

One immediate consequence of the translation is that it is obvious that SWRL rules can entirely replace the role axioms of  $\mathcal{SH}$ :

Axiom	FOL
$Trans(P)$	$\forall x, y, z (P(x, y) \wedge P(y, z) \rightarrow P(x, z))$
$P(X, Y) \wedge P(Y, Z) \rightarrow P(X, Z)$	$\forall X, Y, Z (P(X, Y) \wedge P(Y, Z) \rightarrow P(X, Z))$
$P \sqsubseteq Q$	$\forall x, y (P(x, y) \rightarrow Q(x, y))$
$P(X, Y) \rightarrow Q(X, Y)$	$\forall X, Y (P(X, Y) \rightarrow Q(X, Y))$

Clearly, replacing the variables in the first order translation of the  $\mathcal{SH}$  axioms with their uppercased versions results in the first order translation of the SWRL rule. Thus  $\mathcal{SH}+\text{SWRL}$  is trivially reducible to  $\mathcal{ALC}+\text{SWRL}$ . Of course, concept inclusion axioms may be encoded as SWRL rules analogously to how property inclusions are (at least, if we allow concept expressions as the functor of atoms; otherwise, we need at least some definitions). If, as the SWRL submission does, we permit SWRL rules with empty antecedents, then we can assimilate the ABox axioms (i.e., type, property, and (in)equality assertions) as well.

These reductions, as observed in [13], highlight the fact that Description Logics can be seen as rule languages, given a sufficiently liberal notion of rule. The flipside is that SWRL rules have more in common with Description Logic axioms than with Datalog rules and certainly more than with production or ECA rules.

While we know of no user study comparing the usability of explicit rules to Description Logic style variable free axioms (and we believe that this would be a very difficult study to conduct), there is no shortage of claims that one style or the other is more usable, transparent, or easy to learn. For example,

some researchers working on controlled natural languages (CNLs) for OWL argue that it is better to produce a CNL sentence corresponding to the SWRL rule (including explicit variables!) for transitivity instead of using the more succinct “Property P is transitive.” At least one trade off seems clear: Description Logic “variable free” style is less cluttered with (clearly redundant!) variables, whereas SWRL style is probably initially more comfortable for people familiar with rule languages such as Prolog or with first order logic or with query languages like SQL. The lack of “visual noise” for standard Description Logic axioms (i.e., concept and property subsumption, transitivity, inverse, etc.) seems decisive except for explication purposes. Similarly, the operator style syntax of Description Logics makes it easier to manage complex nested expressions. This suggests that there may be user advantages to continuing to adopt more expressive role constructors (such as the expanded role composition operator in OWL 1.1).

The undecidability of  $\mathcal{SHOIN}$ +SWRL can easily be seen by the fact that it is possible to encode the transitivity of an otherwise simple role using a SWRL rule. Transitive roles in number restrictions are a well known source of undecidability for  $\mathcal{SHIQ}$  (e.g., see [14]), and thus of OWL DL as well. Of course, it is also easy to directly encode undecidable domino tiling problems in SWRL (as described in [13], for which  $\mathcal{SH}$ +SWRL is sufficient) or other known undecidable extensions (such as role-value-maps).

We have seen that we can see SWRL as a very straightforward generalization of description logics. The addition of arbitrary variables to conditionals brings in a lot of expressive power. However, SWRL is a somewhat strange fragment of first order logic. It lacks, at least, n-ary predications and function symbols, but it nevertheless is only semi-decidable. Of course, pure Prolog is another case of an undecidable fragment of first order logic, but there the question of “Why not move to full first order logic?” has a well known answer: Prolog has a proof procedure which gives rise to a powerful, reasonably efficient in the common case, and understandable performance model for computations. In fact various compromises (such as omitting the occurs check) have well understood performance (and soundness/completeness) tradeoffs. Unfortunately, there is no such body of knowledge about SWRL yet.

## 2.2 Definition of DL-Safe SWRL

The free syntax of SWRL (e.g., conjunctions in the head and arbitrary concept expressions as atoms) helps emphasize the similarities of SWRL rules with Description Logic axioms, but it obscures the relation between SWRL rules and rules systems based on Horn clauses, such as Datalog. However, it is easy enough to transform an arbitrary SWRL knowledge base (consisting of a rules part and a Description Logic part) into a form where all the SWRL rules consequents contain only a single conjunct and all the atoms are atomic. (Essentially, definitions are added to the description logic part that replace each complex class atom with a fresh atomic atom, and conjunctive consequences

are eliminated via the Lloyd-Topor transformations [24] as mentioned in the SWRL submission.)

SWRL ontologies in this form of a knowledge base resemble hybrid Description logic and Datalog knowledge bases. By adjusting the rules portion to be semantically more like Datalog rules (that is, so that the rules act only on the explicit facts in the knowledge and other rules), the hybrid flavor becomes stronger including regained decidability. In the translation to first order logic, one can impose the safety restriction (of variables to named individuals) by means of a special DL predicate,  $O$ , which is true of all named individuals (i.e., there is a fact of the form  $O(a)$  for all individual names,  $a$ , appearing in the ontology) and does not appear in any concept expression of the ontology (and thus not in the scope of any existential quantifier).<sup>9</sup> Then, for each variable in the DL-safe rule, an  $O$ -atom with that variable is added to the body.

That is, we extend the translation to first order in the following way. Let  $\Omega(K) = c_1 \dots c_n$  be the set of individual names appearing in a knowledge base  $K$ .  $K' = K \cup \{c_1 : O \dots c_n : O\}$ .  $\Pi$  is then applied to the elements of  $K'$ . Furthermore, let  $\Psi_{dls} = \alpha_1 \wedge \dots \wedge \alpha_n \rightarrow \alpha_{n+1}$  be a DL-Safe rule such that every  $\alpha$  is atomic and  $Var(\Psi_{dls}) = v_1 \dots v_k$  are all the SWRL variables in the atoms  $\alpha_1 \dots \alpha_{n+1}$ . Then:

Axiom Translation

$$\pi(\Psi_{dls}) \forall v_1 \dots v_k (\pi(\alpha_1) \wedge \dots \wedge \pi(\alpha_n) \wedge O(v_1) \wedge \dots \wedge O(v_k) \rightarrow \pi(\alpha_{n+1}))$$

It is clear from the translation that the DL-Safe version of the SWRL rule is much weaker than the unrestricted version. Consider the following simple example (adapted from [39]):

$$\text{Foot} \sqsubseteq \exists \text{partOf}.\text{Leg} \quad (1)$$

$$\text{BurnOnFoot} \sqsubseteq \exists \text{locatedIn}.\text{Foot} \quad (2)$$

$$\text{LegInjury} \equiv \exists \text{locatedIn}.\text{Leg} \quad (3)$$

$$\langle \text{burn1}, \text{foot1} \rangle : \text{locatedIn} \quad (4)$$

$$\langle \text{foot1}, \text{leg1} \rangle : \text{partOf} \quad (5)$$

$$\text{leg1} : \text{Leg} \quad (6)$$

$$\text{foot1} : \text{Foot} \quad (7)$$

$$\text{burn2} : \text{BurnOnFoot} \quad (8)$$

$$\text{locatedIn}(X, Y) \wedge \text{partOf}(Y, Z) \rightarrow \text{locatedIn}(X, Z) \quad (9)$$

If we interpret (8) as an unrestricted SWRL rule, then we can conclude:

$$\langle \text{burn1}, \text{leg1} \rangle : \text{locatedIn} \quad (10)$$

$$\text{burn1} : \text{LegInjury} \quad (11)$$

$$\text{BurnOnFoot} \sqsubseteq \text{LegInjury} \quad (12)$$

$$\text{burn2} : \text{LegInjury} \quad (13)$$

<sup>9</sup> Thanks to an anonymous reviewer for some clarifications on this point.

whereas, if we impose DL-Safety on (8), we can only conclude (10) and (11). Clearly, with DL-Safety, rules can still interact with Description Logic axioms, but only through new ground facts. It is because we infer (10) (via (9)) that we can conclude (11) (via the class definition (3)). Thus, DL-Safety constrains rules to work on and through ground facts.

DL-Safe SWRL rules, in addition to being much more computationally reasonable, may be more cognitively adequate, perhaps *due* to their relative expressive weakness. For users coming from a database background, they can be seen as a data manipulation language. It is at least plausible that it is somewhat easier for the typical user to understand why a missing fact blocks an entailment, than to understand why a subsumption fails because of there only being necessary, but not sufficient, conditions defined.

### 2.3 Built-ins

This more programmatic feel is enhanced by the presence of built-in atoms, that is, atoms with a fixed, predefined interpretation. The SWRL submission includes built-ins for value comparison, mathematics, and string manipulation among others. There are several issues with built-ins with perhaps the most prominent being how they are to be interpreted if their variables are under-instantiated when being evaluated (that is, whether they should be interpreted as arbitrary constraints or more procedurally, e.g., “throwing” an error when a binding is of an inappropriate type or must be drawn arbitrarily from the domain rather than determined by the knowledge base). However, SWRL built-ins seem fairly popular: at least they are reported as desired, as a supplement to OWL’s datatype facility. In spite of this clamor, it is unclear what the right semantics of built-ins (or even just data property atoms) should be, in part due to the fact that the most natural reading of data property atoms (and built-ins) is as general constraints, whereas many users talk about built-ins as if they were procedural attachments.

## 3 Rule Fragments of OWL

### 3.1 Description Logic Programs

Description Logic Programs (DLP) are a naive *Horn fragment* of OWL. They inherit their semantics from OWL, thus adhering to the open world assumption. At the same time, DLPs can be transformed *syntactically* into Logic Programming syntax, and thus provide a kind of basic interoperability between OWL and Logic Programming. Let it be noted, though, that DLP is *not a common* fragment of OWL and Logic Programming, because the two semantics normally considered are different. Nevertheless, the two semantics have a clearly understood and strong relationship, about which we will talk later in Section 3.2.



Compared to OWL DL, DLP is a rather primitive ontology language. It strictly contains, however, the OWL DL fragment of RDFS (see Chapter 3) and has the pleasing property of having polynomial data complexity. It is thus one of the tractable OWL fragments<sup>10</sup> discussed within the currently ongoing OWL 2 standardization effort.<sup>11</sup>

According to an analysis in [41], most of the existing OWL ontologies are almost completely contained in DLP.<sup>12</sup> An example is the Semantic Web Research Community ontology, SWRC [40],<sup>13</sup> which is the most imported ontology on the web.<sup>14</sup>

DLP was originally presented in [11], and a thorough treatment can be found in [41].

### Definition of DLP

Originally, DLP was presented as fragment of OWL. We present DLP in an alternative and more constructive way, which exposes the modeling capabilities and the limitations of it. It can be considered a kind of *normal form* for Description Logic Programs.

We need to fix terminology first. We understand DLP as a *semantic* fragment of OWL, i.e. we abstract (for the time being) from a concrete syntax: Every OWL statement which is *semantically* equivalent — in the sense of first order logic — to a (finite) set of function-free Horn clauses constitutes a valid DLP statement.<sup>15</sup> Allowing integrity constraints, we call the resulting fragment DLP IC (or just IC). Allowing integrity constraints and equality, we call the resulting fragment DLP ICE (or ICE). We write DLP<sup>+</sup> for the (semantic) fragment common to OWL DL and (function-free non-disjunctive) Datalog. Analogously, we write DLP<sup>+</sup> IC, IC<sup>+</sup>, etc.

Now to the definition. Allowed are the following, where  $a, b, a_i$  stand for individuals,  $C$  stands for a concept name and  $R, Q, R_i, Q_{i,j}$  stand for role names.

<sup>10</sup> <http://www.w3.org/Submission/2006/SUBM-owl11-tractable-20061219/>

<sup>11</sup> <http://www.w3.org/2007/OWL/>

<sup>12</sup> We would like to mention that it has been argued whether this analysis is appropriate.

<sup>13</sup> <http://www.aifb.uni-karlsruhe.de/about.html>

<sup>14</sup> <http://ebiquity.umbc.edu/blogger/2007/06/15/how-owlimport-is-used/>

<sup>15</sup> In our terminology, the set of OWL Lite statements  $\{C \sqsubseteq D \sqcup E, D \equiv E\}$  would not qualify as a set of DLP statements, although it is semantically equivalent to  $\{C \sqsubseteq D, D \equiv E\}$ , which is expressible in DLP. We are well aware of this restriction, but will not be concerned with it in the moment, because this more general notion of semantic equivalence is not readily accessible by syntactic means. Note, however, that  $C \sqsubseteq D \sqcup D$  qualifies as a DLP statement, since it is semantically equivalent to  $C \sqsubseteq D$ .

- ABox:
  - $C(a)$  (individual assertion)
  - $R(a, b)$  (property assertion)
  - $a = b$  (ICE) (individual equivalence)
- Property Characteristics:
  - $R \equiv Q$  (equivalence)
  - $R \sqsubseteq Q$  (subproperty)
  - $\top \sqsubseteq \forall R.C$  ( $C \neq \perp$ ) (domain)
  - $\top \sqsubseteq \forall R^-.C$  ( $C \neq \perp$ ) (range)
  - $R \equiv Q^-$  (inverse)
  - $R \equiv R^-$  (symmetry)
  - $\top \sqsubseteq \leq 1R$  (ICE) (functionality)
  - $\top \sqsubseteq \leq 1R^-$  (ICE) (inverseFunctionality)
- TBox: We allow expressions of the form

$$\exists Q_{1,1}^{(-)} \dots \exists Q_{1,m_1}^{(-)}. \mathbf{Left}_1 \sqcap \dots \sqcap \exists Q_{k,1}^{(-)} \dots \exists Q_{k,m_k}^{(-)}. \mathbf{Left}_k \sqsubseteq \forall R_1^{(-)} \dots \forall R_n^{(-)}. \mathbf{Right}$$

where the following apply.

- For DLP we allow  $\mathbf{Left}_j$  to be of the forms  $C$ ,  $\{o_1, \dots, o_n\}$ ,  $\perp$  or  $\top$ , and  $\mathbf{Right}$  to be of the forms  $C$  or  $\top$ .
- For DLP IC we allow  $\mathbf{Left}_j$  to be of the forms  $C$ ,  $\{o_1, \dots, o_n\}$ ,  $\perp$ , or  $\top$ , and  $\mathbf{Right}$  to be of the form  $C$ ,  $\top$ , or  $\perp$ .
- For DLP ICE we allow  $\mathbf{Left}_j$  to be of the forms  $C$ ,  $\{o_1, \dots, o_n\}$ ,  $\perp$ , or  $\top$ , and  $\mathbf{Right}$  to be of the form  $C$ ,  $\top$ ,  $\perp$ , or  $\{o\}$ .
- For the DLP<sup>+</sup> versions we furthermore allow  $\mathbf{Right}$  to be of the form  $\exists R^{(-)}. \{a\}$ .

The superscript  $(-)$  shall indicate that an inverse symbol may occur in these places. Note that (by a common abuse of notation) we allow any of  $k, m_i, n$  to be zero. For  $k = 0$  the left hand side becomes  $\top$ . Note also that we could have disallowed  $\perp$  on the left and  $\top$  on the right, since in either case the statement becomes void. Likewise, it would suffice to require  $n = 0$  in all cases, since universal quantifiers on the right are expressible using existentials on the left. Disallowing the existential quantifiers on the left (while keeping universals on the right) is also possible, but at the expense of the introduction of an abundance of new concept names. As an example, note that  $\exists R.C \sqcap \exists Q.D \sqsubseteq E$  would have to be translated into the set of statements  $\{C_1 \sqcap D_1 \sqsubseteq E, C \sqsubseteq \forall R^-.C_1, D \sqsubseteq \forall Q^-.D_1\}$ , where  $C_1$  and  $D_1$  are new concept names. Our representation is more compact.

Using any of the established syntaxes of the OWL language, an OWL axiom is said to be in DLP if its translation into Description Logic syntax results in a finite set of statements of the above mentioned form.

### An Example

We give a small example ontology which displays the modeling expressivity of DLP.

For the TBox, we model the following sentences.

- (1) Every man or woman is an adult.
- (2) A grown-up is a human who is an adult.
- (3) A woman who has somebody as a child, is a mother.
- (4) An orphan is the child of humans who are dead.
- (5) A lonely child has no siblings.
- (6) AIFB researchers are employed by the University of Karlsruhe.

They can be written in DL syntax as follows – the axioms actually constitute an  $\mathcal{ALC}\mathcal{IO}$  TBox (see Chapter 1).

$$\text{Man} \sqcup \text{Woman} \sqsubseteq \text{Adult} \quad (14)$$

$$\text{GrownUp} \sqsubseteq \text{Human} \sqcap \text{Adult} \quad (15)$$

$$\text{Woman} \sqcap \exists \text{childOf}^- . \top \sqsubseteq \text{Mother} \quad (16)$$

$$\text{Orphan} \sqsubseteq \forall \text{childOf} . (\text{Dead} \sqcap \text{Human}) \quad (17)$$

$$\text{LonelyChild} \sqsubseteq \neg \exists \text{siblingOf} . \top \quad (18)$$

$$\text{AIFBResearcher} \sqsubseteq \exists \text{employedBy} . \{\text{UKARL}\} \quad (19)$$

Using the forms of DLP statements which we introduced, the TBox can be written as follows.

$$\text{Man} \sqsubseteq \text{Adult} \quad (1)$$

$$\text{Woman} \sqsubseteq \text{Adult} \quad (1)$$

$$\text{GrownUp} \sqsubseteq \text{Human} \quad (2)$$

$$\text{GrownUp} \sqsubseteq \text{Adult} \quad (2)$$

$$\text{Woman} \sqcap \exists \text{childOf}^- . \top \sqsubseteq \text{Mother} \quad (3)$$

$$\text{Orphan} \sqsubseteq \forall \text{childOf} . \text{Dead} \quad (4)$$

$$\text{Orphan} \sqsubseteq \forall \text{childOf} . \text{Human} \quad (4)$$

$$\text{LonelyChild} \sqsubseteq \forall \text{siblingOf} . \perp \quad (5)$$

$$\text{AIFBResearcher} \sqsubseteq \exists \text{employedBy} . \{\text{UKARL}\} \quad (6)$$

We note that for (5) we require DLP IC, while for (6) we require DLP<sup>+</sup>. As an example for an RBox, we use the following.

$\text{parentOf} \equiv \text{childOf}^-$	parentOf and childOf are inverse roles.
$\text{parentOf} \sqsubseteq \text{ancestorOf}$	parentOf is a subrole of ancestorOf.
$\text{fatherOf} \sqsubseteq \text{parentOf}$	fatherOf is a subrole of parentOf.
$\top \sqsubseteq \forall \text{ancestorOf} . \text{Human}$	Human is the domain of ancestorOf.
$\top \sqsubseteq \leq 1 \text{fatherOf}^-$	fatherOf is inverse functional.

We can populate the classes and roles by means of an ABox e.g. in the following way.

$$\begin{aligned}
\{\text{Ian, Benjamin, Raphael, Horrocks}\} &\sqsubseteq \text{Man} \\
\{\text{Yue, Ulrike}\} &\sqsubseteq \text{Woman} \\
\text{Ian} &= \text{Horrocks} \\
\langle \text{Ian, UMAN} \rangle &: \text{employedBy} \dots
\end{aligned}$$

Note that an ABox statement such as

$$\{\text{Yue, Ulrike}\} \sqsubseteq \text{Woman}$$

is simply syntactic sugar for the two statements

$$\text{Yue} : \text{Woman} \quad \text{Ulrike} : \text{Woman}.$$

We therefore consider it to be part of the ABox. To be precise, the original statement actually is (syntactically) not in OWL Lite, but the equivalent set of three ABox statements is. The statement  $\text{Ian} = \text{Horrocks}$  requires DLP ICE.

Note also that class inclusions cannot in general be replaced by equivalences. For example, the statement

$$\text{Adult} \sqsubseteq \text{Man} \sqcup \text{Woman}$$

is not in DLP.

### Relation to Logic Programming

A DLP knowledge base can be translated into Horn logic, and the latter can be expressed using Logic Programming syntax. To be more precise, DLP translates syntactically into Datalog. Let us first continue our example, giving the knowledge base in Datalog form.

The TBox is as follows.

$$\text{Adult}(y) \leftarrow \text{Man}(y) \tag{1}$$

$$\text{Adult}(y) \leftarrow \text{Woman}(y) \tag{1}$$

$$\text{Human}(y) \leftarrow \text{GrownUp}(y) \tag{2}$$

$$\text{Adult}(y) \leftarrow \text{GrownUp}(y) \tag{2}$$

$$\text{Mother}(y) \leftarrow \text{childOf}(x, y) \wedge \text{Woman}(y) \tag{3}$$

$$\text{Dead}(y) \leftarrow \text{Orphan}(x) \wedge \text{childOf}(x, y) \tag{4}$$

$$\text{Human}(y) \leftarrow \text{Orphan}(x) \wedge \text{childOf}(x, y) \tag{4}$$

$$\leftarrow \text{LonelyChild}(x) \wedge \text{siblingOf}(x, y) \tag{5}$$

$$y = \text{UKARL} \leftarrow \text{AIFBResearcher}(x) \wedge \text{employedBy}(x, y) \tag{6}$$

Translating the RBox yields the following statements.

$$\begin{aligned}
\text{parentOf}(x, y) &\leftarrow \text{childOf}(y, x) \\
\text{childOf}(x, y) &\leftarrow \text{childOf}(y, x) \\
\text{ancestorOf}(x, y) &\leftarrow \text{parentOf}(x, y) \\
\text{parentOf}(x, y) &\leftarrow \text{fatherOf}(x, y) \\
\text{Human}(y) &\leftarrow \text{ancestorOf}(x, y) \\
y = z &\leftarrow \text{fatherOf}(y, x) \wedge \text{fatherOf}(z, x)
\end{aligned}$$

Translated as such, DLP knowledge bases can also be evaluated under Datalog semantics, which differs from the OWL semantics. The two semantics, however, coincide on the set of inferred instances of named classes, i.e. ABox reasoning in DLP can be done using Datalog semantics. The formal relationship between the two semantics is as follows.

**Theorem 1.** *Let  $K$  be a DLP knowledge base and let  $K'$  be the translation of  $K$  into Datalog syntax. Let  $C$  be a named class and  $a$  be a named individual. Then  $K \models_{\text{OWL}} C(a)$  under the OWL semantics iff  $K \models_{\text{Datalog}} C(a)$  under the Datalog semantics.<sup>16</sup>*

Very recently, in the wake of the already mentioned forthcoming revision of the OWL standard, which will be based on the *SR<sub>OIQ</sub>* description logic, a naive rule fragment of *SR<sub>OIQ</sub>* has been identified which considerably extends DLP, but is not a fragment of *SH<sub>OIN</sub>(D)* (i.e. of OWL DL). It is called *SR<sub>OIQ</sub> Rules*, and the interested reader shall be pointed to [21] (see also [10, 36]).

### 3.2 Horn-*SHIQ*

Horn-*SHIQ* is another Horn fragment of OWL DL, which encompasses DLP. To be precise, Horn-*SHIQ* has a feature which lies outside OWL DL, namely the use of qualified number restrictions.<sup>17</sup> The corresponding fragment lying within OWL DL would be Horn-*SHIN*, which is obtained from Horn-*SHIQ* by simply disallowing the use of qualified number restrictions.

Horn-*SHIQ* has the pleasing property that it is of tractable (i.e. polynomial) data complexity [15, 29], while its combined complexity is ExpTime [20]. It thus provides striking balance between expressivity and ABox reasoning scalability. While Horn-*SHIQ* can be dealt with by any of the standard

<sup>16</sup> Note that – since we have non-disjunctive rules – the only effect of integrity constraints is that they can render the knowledge base to be inconsistent. Equality also needs some explanations: The theorem assumes that we do not use the unique name assumption, and equality thus has the same meaning as it has under OWL DL, e.g. two different constants can be equal, meaning that they denote the same individual.

<sup>17</sup> However, Horn-*SHIQ* lies within the proposed OWL 2 language, see <http://www.w3.org/Submission/owl11-tractable/>

$$\begin{aligned}
\mathbf{C}_1^+ &\leftarrow \top \mid \perp \mid \neg \mathbf{C}_1^- \mid \mathbf{C}_1^+ \sqcap \mathbf{C}_1^+ \mid \mathbf{C}_0^+ \sqcup \mathbf{C}_1^+ \mid \exists \mathbf{R}.\mathbf{C}_1^+ \mid \forall \mathbf{S}.\mathbf{C}_1^+ \mid \forall \mathbf{R}.\mathbf{C}_0^+ \mid \\
&\quad \mid \geq n \mathbf{R}.\mathbf{C}_1^+ \mid \leq 1 \mathbf{R}.\mathbf{C}_0^- \mid \mathbf{A} \\
\mathbf{C}_1^- &\leftarrow \top \mid \perp \mid \neg \mathbf{C}_1^+ \mid \mathbf{C}_0^- \sqcap \mathbf{C}_1^- \mid \mathbf{C}_1^- \sqcup \mathbf{C}_1^- \mid \exists \mathbf{S}.\mathbf{C}_1^- \mid \exists \mathbf{R}.\mathbf{C}_0^- \mid \forall \mathbf{R}.\mathbf{C}_1^- \mid \\
&\quad \mid \geq 2 \mathbf{R}.\mathbf{C}_0^- \mid \leq n \mathbf{R}.\mathbf{C}_1^+ \mid \mathbf{A} \\
\mathbf{C}_0^+ &\leftarrow \top \mid \perp \mid \neg \mathbf{C}_0^- \mid \mathbf{C}_0^+ \sqcap \mathbf{C}_0^+ \mid \mathbf{C}_0^+ \sqcup \mathbf{C}_0^+ \mid \forall \mathbf{R}.\mathbf{C}_0^+ \\
\mathbf{C}_0^- &\leftarrow \top \mid \perp \mid \neg \mathbf{C}_0^+ \mid \mathbf{C}_0^- \sqcap \mathbf{C}_0^- \mid \mathbf{C}_0^- \sqcup \mathbf{C}_0^- \mid \exists \mathbf{R}.\mathbf{C}_0^- \mid \mathbf{A}
\end{aligned}$$

**Table 1.** A grammar for defining Horn-*SHIQ*.  $\mathbf{A}$ ,  $\mathbf{R}$ , and  $\mathbf{S}$  denote the sets of all concept names, role names, and simple role names, respectively. The presentation is slightly simplified by exploiting associativity and commutativity of  $\sqcap$  and  $\sqcup$ , and by omitting  $\geq 1 R.C$  if  $\exists R.C$  is present. The grammar for Horn-*SHIN* is obtained by replacing the qualifying class by  $\top$  in all number restrictions.

OWL reasoners, it is specifically supported by KAON2<sup>18</sup> [29] – see Chapter 24 – and by the new HermiT system<sup>19</sup> [34].

The original definition of Horn-*SHIQ*, due to [15, 29], remained implicit. It was defined as the fragment of *SHIQ* which, after transformation by the KAON2 algorithms, resulted in (non-disjunctive) Datalog. It is outside the scope of this chapter to detail the underlying KAON2 algorithms, and we refer the reader to Chapter 24 and to [29] for this. Instead, we give an alternative definition by means of a grammar.

### Definition of Horn-*SHIQ*

The following definition is taken from [19], where also a formal proof can be found that this definition coincides with the original notion.

We say that a *SHIQ* axiom  $C \sqsubseteq D$  is *Horn* if the concept expression  $\neg C \sqcup D$  has the form  $\mathbf{C}_1^+$  as defined by the context-free grammar in Table 1. A *SHIQ* knowledge base with an extensionally reduced ABox is *in Horn-*SHIQ** if all of its TBox axioms are Horn.

It is easily seen by referring to the definition on page 9, that DLP IC is indeed contained in Horn-*SHIQ*. Just note that the use of  $\{o_1, \dots, o_n\}$  on the **Left** is removed by extensionally reducing the ABox. Intuitively speaking, Horn-*SHIQ* adds the free use of role restrictions to DLP, as e.g. existential restriction can be used freely.

### An Example

As an example for a Horn-*SHIQ* knowledge base, consider the ontology in Table 2, which exemplifies the expressivity possible in Horn-*SHIQ*. Note in particular the free use of role restrictions which is not possible in DLP.

<sup>18</sup> <http://kaon2.semanticweb.org/>

<sup>19</sup> <http://www.cs.man.ac.uk/~bmotik/Hermit/>

<b>TBox/RBox</b>	
(1)	Parent $\equiv \exists \text{ hasChild.}\top$
(2)	Person $\sqsubseteq \exists \text{ childOf. Person}$
(3)	ManyChildren $\sqsubseteq \geq 2 \text{ hasChild.}\top$
(4)	NoSiblings $\sqsubseteq \text{ Person} \sqcap \forall \text{ childOf. } (\leq 1 \text{ hasChild.}\top)$
(5)	childOf $\equiv \text{ hasChild}^{-1}$

<b>ABox</b>	
< Elaine, Sir Lancelot >	: hasChild
Lancelot du Lac	: noSiblings
< Lancelot du Lac, Elaine >	: childOf

**Table 2.** An example ontology in Horn-*SHIQ*.

### Relation to Logic Programming

Horn-*SHIQ* can be translated into Datalog syntax, i.e. it can be understood as a rule fragment of *SHIQ*. We do not have the space to detail the underlying KAON2 translation algorithms, and refer to Chapter 24 and [29] for this. It shall be noted, though, that this transformation is *not* an equivalence translation in the sense that the original ontology and its translation have the same models. The relationship is more intricate, as given in the following theorem from [29].

**Theorem 2.** *Let  $K$  be a Horn-*SHIQ* knowledge base and let  $D(K)$  be the transformation of  $K$  into Datalog syntax resulting from the application of the KAON2 transformation algorithms. Then the following hold.*

1.  $K$  is unsatisfiable if and only if  $D(K)$  is unsatisfiable.
2.  $K \models \alpha$  if and only if  $D(K)$  entails  $\alpha$  under Datalog semantics, where  $\alpha$  is of the form  $A(a)$  or  $R(a, b)$ , and  $A$  is an atomic concept.
3.  $K \models C(a)$  for a non-atomic concept  $C$  if and only if, for  $Q$  a new atomic concept,  $Q(a)$  is entailed by  $D(K \cup \{C \sqsubseteq Q\})$  under the Datalog semantics.

In order to exemplify the mentioned transformation, we give a translation of the ontology from Table 2 into a logic program which can be executed under Prolog with tabling.<sup>20</sup> This example is due to [37].

So consider the ontology given in Table 2. The corresponding translation to logic programming is given in Table 3. Let us first consider the upper part, which shows the rules directly created in the translation. Some of the rules clearly represent (part of) some *SHIQ*-axiom, as is the case for “person( $X$ )  $\leftarrow$  nosiblings( $X$ ).” and axiom (4). Other rules are obtained by more complicated reasoning steps, such as e.g. “parent( $X$ )  $\leftarrow$  manychildren( $X$ ).” which is obtained from axioms (3) and (1). While such rules are still fairly self-explanatory, there are also a number of axioms that include predicates of the

<sup>20</sup> Using e.g. XSB-Prolog, <http://xsb.sourceforge.net/>.

---

$\text{person}(X) \leftarrow \text{nosiblings}(X).$ $\text{person}(X_{f_3}) \leftarrow \text{person}(X), S_{f_3}(X, X_{f_3}).$ $\text{parent}(X) \leftarrow \text{haschild}(X, Y).$ $\text{parent}(X) \leftarrow \text{manychildren}(X).$ $\text{haschild}(Y, X) \leftarrow \text{childof}(X, Y).$ $\text{haschild}(X, X_{f_1}) \leftarrow \text{manychildren}(X), S_{f_1}(X, X_{f_1}).$ $\text{haschild}(X, X_{f_2}) \leftarrow \text{parent}(X), S_{f_2}(X, X_{f_2}).$ $\text{haschild}(X, X_{f_0}) \leftarrow \text{manychildren}(X), S_{f_0}(X, X_{f_0}).$ $\text{childof}(X, X_{f_3}) \leftarrow \text{person}(X), S_{f_3}(X, X_{f_3}).$ $\text{childof}(Y, X) \leftarrow \text{haschild}(X, Y).$
$Y_1 \approx Y_2 \leftarrow \text{nosiblings}(X), \text{childof}(X, Z), \text{haschild}(Z, Y_1), \text{haschild}(Z, Y_2).$ $\text{inc} \leftarrow \text{manychildren}(X), \text{nosiblings}(X_0), \text{childof}(X_0, X).$ $\text{inc} \leftarrow X_{f_1} \approx X_{f_0}, \text{manychildren}(X), S_{f_1}(X, X_{f_1}), S_{f_0}(X, X_{f_0}).$
<hr/> $S_f(X, f(X)) \leftarrow \text{O}(X).$ $\text{HU}(X) \leftarrow \text{O}(X).$ $\text{HU}(f(X)) \leftarrow \text{O}(X). \quad (\text{for } f \in \{f_0, f_1, f_2, f_3\})$ $X \approx X \leftarrow \text{HU}(X).$ $X \approx Y \leftarrow Y \approx X, \text{HU}(X), \text{HU}(Y).$ $X \approx Z \leftarrow X \approx Y, Y \approx Z, \text{HU}(X), \text{HU}(Y), \text{HU}(Z).$ $C(Y) \leftarrow C(X), X \approx Y, \text{HU}(X), \text{HU}(Y).$ <div style="text-align: right; padding-right: 20px;"><math>(\text{for } C \in \{\text{person}, \text{parent}, \text{manychildren}, \text{nosiblings}\})</math></div> $R(Y_1, Y_2) \leftarrow R(X_1, X_2), X_1 \approx Y_1, X_2 \approx Y_2, \text{HU}(X_1), \text{HU}(X_2), \text{HU}(Y_1), \text{HU}(Y_2).$ <div style="text-align: right; padding-right: 20px;"><math>(\text{for } R \in \{\text{childof}, \text{haschild}\})</math></div> <hr/> $\text{O}(\text{Elaine}). \quad \text{O}(\text{Sir Lancelot}). \quad \text{O}(\text{Lancelot du Lac}).$

**Table 3.** Continuation of the example from Table 2: Its translation into Horn-logic, consisting of the translated rules (middle) and auxiliary axioms (bottom).

form  $S_f(X, X_f)$  which do not appear in the original knowledge base. These predicates are introduced during the transformation process, more precisely during a step for eliminating Skolem-functions. Intuitively,  $S_f(X, Y)$  holds if and only if  $Y = f(X)$ . However, the predicates  $S_f$  are only satisfied for a finite number of constants, since arbitrary application of functions is not needed and might even lead to undecidability. The exact number of additional function symbols may vary from case to case. Finally, two of the rules represent integrity constraints by means of the predicate  $\text{inc}$ , in the sense that  $\text{inc}$  evaluates to **true** if the integrity constraint is violated.

The rules in the lower part of Table 3 define various auxiliary predicates that are needed for the correctness of the translation. In order to restrict these definitions to a finite number of terms, we introduce a predicate  $\text{O}$  that specifies the individuals for which the program applies. In our case, these are just the individuals from the ABox. Using  $\text{O}$ , we define  $S_f$  as discussed above. Further, we introduce a predicate  $\text{HU}$  defining which terms are considered in the program, namely individuals from  $\text{O}$  and their immediate successors for



each function symbol. The remaining rules yield a necessary equality theory, restricted to the terms in HU.

The resulting program now allows us to conclude several ABox statements. For example, we can derive that “parent(Elaine)” and that “Sir Lancelot  $\approx$  Lancelot du Lac.”

## 4 Hybrid Approaches

In Section 2, we discussed the SWRL rule extension of OWL, which basically follows the design principles of OWL DL by adhering to the open world assumption and by being semantically based on first-order predicate logic. In Section 3, we discussed rule languages which are fragments of OWL DL, and thus inherit its semantics.

We have focussed on these perspectives for several reasons.

- They align with OWL DL semantically in a very natural way.
- They are supported by some of the most prominent and powerful OWL reasoning engines.
- They appear to be least disputed as to their importance for ontology modeling.

At the same time, there is a multitude of proposals for *hybrid* systems which comprise both classical OWL reasoning and traditional rule-based approaches like logic programming in different variants. The quest for such hybrid solutions is currently still ongoing, but is not yet close to a conclusion. We briefly discuss the two approaches which we consider to be most mature at this stage.

The first of these is called Hybrid MKNF knowledge bases due to [30, 31], which roughly is an autoepistemic extension of OWL DL with DL-safe rules. The result is a seamless integration of open- and closed-world reasoning within a single framework which encompasses both OWL DL reasoning and prominent forms of non-monotonic reasoning. Recent investigations [17, 18] strive at establishing an implementable semantics for this approach.

The second approach is based on an integration of OWL DL reasoning with Answer Set Programming, which has been realised as the dlvhx system [9, 8, 38]. The integration is less strong as for Hybrid MKNF knowledge bases, and basically consists of two reasoning engines which interact in a bidirectional way when reasoning over knowledge bases.

Many additional approaches are currently being investigated and proposed, ranging from tightly integrated ones to loosely coupled systems. We list recent references as a starting point for the interested reader [23, 4, 5, 27, 7, 28, 25, 35, 12].

## 5 Conclusions

Our discussion of ontologies and rules is centered on two specific paradigms of “ontology” formalisms and rule formalisms: Description Logics and logic programming (though we have only lightly touched on issues with various forms of non-monotonicity and similar core features of logic programming systems). While arguably these are both very prominent, one might say dominant, formalisms in the ontology engineering communities today — and there is an enormous amount of work on their combination as we have seen — historically, this was not always the case. For example, if we consider expert systems of the 1970s and 1980s such as Mycin [3] we find that production rule languages were quite prominent as rule formalisms for ontologies, and for knowledge representation more generally. Today, that community generally represents itself as dealing with “business rules” almost exclusively<sup>21</sup>. This is reflected in the make up of the currently ongoing Rules Interchange Format (RIF) working group<sup>22</sup> of the W3C.

Integrating *logic programming* and business rules, much less description logics, has proven to be challenging. This is rather surprising given the obvious parallels between Prolog and OPS5 style rules and the natural thought that use of forward chaining inference methods — or more specifically variants of the Rete algorithm — is irrelevant to the semantics of the rules. It seems *very* natural to think that a (simple) relational database with some (simple) Event-Condition-Action (ECA) rules is equivalent to that relational database with some Datalog rules. However, a database administrator may have reason to prefer that the ECA rule evaluation *modified the database*. In fact, that may have been a key feature of those rules: consider a situation where one wished to validate certain inputs but *only at input time*, that is, subsequent operations are licensed to violate the validation criteria. In such contexts, the notion of “assert” and “retract” have representational significance. If you add, in the action language, the ability to execute code fragments written in a programming language, it seems clear that there is a fundamental divergence. Unfortunately, unlike with Description Logics and logic programming, we do not currently have a well established common semantic framework (i.e., modal theory) ready to hand to aid us with integration.

## References

1. Patrick Blackburn, Johan F. A. K. van Benthem, and Frank Wolter. *Handbook of Modal Logic, Volume 3 (Studies in Logic and Practical Reasoning)*. Elsevier Science Inc., New York, NY, USA, 2006.

<sup>21</sup> See <http://www.businessrulesgroup.org/brmanifesto.htm> for a taste of business rule concerns.

<sup>22</sup> <http://www.w3.org/2005/rules/wg>

2. Ronald J. Brachman, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lori A. Resnick. Living with CLASSIC: when and how to use a KL-ONE-like language. In John Sowa, editor, *Principles of semantic networks*. Morgan Kaufmann, San Mateo, US, 1990.
3. Bruce Buchanan and Edward Shortliffe. *Rule-Based Expert Systems: The MYCIN experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, MA, 1984.
4. Jos de Bruijn, Thomas Eiter, Axel Polleres, and Hans Tompits. Embedding non-ground logic programs into autoepistemic logic for knowledge-base combination. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, January 6–12 2007. AAAI Press.
5. Jos de Bruijn, David Pearce, Axel Polleres, and Agustín Valverde. Quantified equilibrium logic and hybrid rules. In Marchiori et al. [26], pages 58–72.
6. Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. AL-log: Integrating datalog and description logics. *Journal of Intelligent Information Systems*, 10(3):227–252, 1998.
7. Włodzimierz Drabent and Jan Maluszynski. Well-founded semantics for hybrid rules. In Marchiori et al. [26], pages 1–15.
8. Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits. dlvhex: A prover for semantic-web reasoning under the answer-set semantics. In *2006 IEEE / WIC / ACM International Conference on Web Intelligence (WI 2006), 18-22 December 2006, Hong Kong, China*, pages 1073–1074. IEEE Computer Society, 2006.
9. Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits. Effective integration of declarative rules with external evaluations for semantic-web reasoning. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings*, volume 4011 of *Lecture Notes in Computer Science*, pages 273–287. Springer, 2006.
10. Francis Gasse, Ulrike Sattler, and Volker Haarslev. Rewriting rules into SROIQ axioms. In *Proc. 21st Int. Workshop on Description Logics (DL-08), Dresden, Germany, 2008*.
11. Benjamin Grosz, Ian Horrocks, Raphael Volz, and Stefan Decker. Description logic programs: Combining logic programs with description logics. In *Proc. of WWW 2003, Budapest, Hungary, May 2003*, pages 48–57. ACM, 2003.
12. Stijn Heymans, Davy Van Nieuwenborgh, and Dirk Vermeir. Nonmonotonic ontological and rule-based reasoning with extended conceptual logic programs. In *Proceedings ESWC2005*, volume 3532 of *Lecture Notes in Computer Science*, pages 392–407. Springer, 2005.
13. Ian Horrocks and Peter F. Patel-Schneider. A proposal for an OWL rules language. In *Proc. of the Thirteenth International World Wide Web Conference (WWW 2004)*, pages 723–731. ACM, 2004.
14. Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.
15. Ullrich Hustadt, Boris Motik, and Ulrike Sattler. Data complexity of reasoning in very expressive description logics. In *Proceedings of the 19th International Joint Conferenc on Artificial Intelligence (IJCAI)*, pages 466–471, 2005.
16. Ullrich Hustadt, Renate A. Schmidt, and Lilia Georgieva. A survey of decidable first-order fragments and description logics. *Journal of Relational Methods in Computer Science*, 1:251–276, 2004. Invited overview paper.

17. Matthias Knorr, Jose Alferes, and Pascal Hitzler. Towards tractable local closed world reasoning for the semantic web. In J. Neves, M. F. Santos, and J. Machado, editors, *Progress in Artificial Intelligence, 13th Portuguese Conference on Artificial Intelligence, EPIA 2007, Guimaraes, Portugal, December 3-7, 2007, Proceedings*, volume 4874 of *Lecture Notes in Computer Science*, pages 3–14. Springer, 2007.
18. Matthias Knorr, Jose Alferes, and Pascal Hitzler. A coherent well-founded model for Hybrid MKNF knowledge bases. In *Proceedings of the 18th European Conference on Artificial Intelligence, ECAI-08, Patras, Greece, 2008*. To appear.
19. Markus Krötzsch, Pascal Hitzler, Denny Vrandečić, and Michael Sintek. How to reason with OWL in a logic programming system. In *Proc. 2nd Int. Conf. on Rules and Rule Markup Languages for the Semantic Web (RuleML 2006)*, Athens, Georgia, USA, 2006. Springer.
20. Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. Complexity of Horn description logics. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence, AAAI-07, Vancouver, British Columbia, Canada, July 2007*, pages 452–457, 2007.
21. Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. Description logic rules. In *Proceedings of the 18th European Conference on Artificial Intelligence, ECAI-08, Patras, Greece, 2008*. To appear.
22. Alon Y. Levy and Marie-Christine Rousset. CARIN: A representation language combining Horn rules and description logics. In *European Conference on Artificial Intelligence*, pages 323–327, 1996.
23. Alon Y. Levy and Marie-Christine Rousset. Combining Horn rules and description logics in CARIN. *Artificial Intelligence*, 104:68–78, 1998.
24. John W. Lloyd. Foundations of logic programming (second, extended edition). In *Springer series in symbolic computation*, New York, 1987. Springer-Verlag.
25. Thomas Lukasiewicz. A novel combination of answer set programming with description logics for the semantic web. In Enrico Franconi, Michael Kifer, and Wolfgang May, editors, *The Semantic Web: Research and Applications, 4th European Semantic Web Conference, ESWC 2007, Innsbruck, Austria, June 3-7, 2007, Proceedings*, volume 4519 of *Lecture Notes in Computer Science*, pages 384–398. Springer, 2007.
26. Massimo Marchiori, Jeff Z. Pan, and Christian de Sainte Marie, editors. *Web Reasoning and Rule Systems, First International Conference, RR 2007, Innsbruck, Austria, June 7-8, 2007, Proceedings*, volume 4524 of *Lecture Notes in Computer Science*. Springer, 2007.
27. Tobias Matzner and Pascal Hitzler. Any-world access to OWL from Prolog. In J. Hertzberg, M. Beetz, and R. Englert, editors, *KI 2007: Advances in Artificial Intelligence, 30th Annual German Conference on AI, KI 2007, Osnabrück, Germany, September 2007, Proceedings*, volume 4667 of *Lecture Notes in Computer Science*, pages 84–98. Springer, 2007.
28. Jing Mei, Zuoquan Lin, and Harold Boley. *LC: An integration of description logic and general rules*. In Marchiori et al. [26], pages 163–177.
29. Boris Motik. *Reasoning in Description Logics using Resolution and Deductive Databases*. PhD thesis, Universität Karlsruhe (TH), Germany, 2006.
30. Boris Motik, Ian Horrocks, Riccardo Rosati, and Ulrike Sattler. Can OWL and logic programming live together happily ever after? In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael

- Uschold, and Lora Aroyo, editors, *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 501–514. Springer, 2006.
31. Boris Motik and Riccardo Rosati. A faithful integration of description logics with logic programming. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 477–482. AAAI Press, 2007.
  32. Boris Motik, Ulrike Sattler, and Rudi Studer. Query answering for OWL-DL with rules. In *International Semantic Web Conference*, pages 549–563, 2004.
  33. Boris Motik, Ulrike Sattler, and Rudi Studer. Query answering for OWL-DL with rules. *Journal of Web Semantics*, 3(1):41–60, 2005.
  34. Boris Motik, Rob Shearer, and Ian Horrocks. Optimized Reasoning in Description Logics using Hypertableaux. In *Proc. of the 21st Conference on Automated Deduction (CADE-21)*, volume 4603 of *Lecture Notes in Artificial Intelligence*, pages 67–83, Bremen, Germany, July 17–20 2007. Springer.
  35. Riccardo Rosati. Tight integration of description logics and disjunctive datalog. In *Proceedings KR2006*, pages 68–78. AAAI Press, 2006.
  36. Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler. All elephants are bigger than all mice. In *Proc. 21st Int. Workshop on Description Logics (DL-08)*, Dresden, Germany, 2008.
  37. Sebastian Rudolph, Markus Krötzsch, Pascal Hitzler, Michael Sintek, and Denny Vrandečić. Efficient OWL reasoning with logic programs – Evaluations. In *Proceedings of the First International Conference on Web Reasoning and Rule Systems, RR2007, Innsbruck, Austria, June 2007*, volume 4524 of *Lecture Notes in Computer Science*, pages 370–373. Springer, 2007.
  38. Roman Schindlauer. *Answer-Set Programming for the Semantic Web*. PhD thesis, Vienna University of Technology, Austria, 2006.
  39. Julian Seidenberg and Alan L. Rector. Representing transitive propagation in OWL. In David W. Embley, Antoni Olivé, and Sudha Ram, editors, *ER*, volume 4215 of *Lecture Notes in Computer Science*, pages 255–266. Springer, 2006.
  40. York Sure, Stephan Bloehdorn, Peter Haase, Jens Hartmann, and Daniel Oberle. The SWRC ontology – Semantic Web for Research Communities. In Carlos Bento, Amílcar Cardoso, and Gael Dias, editors, *Proceedings of the 12th Portuguese Conference on Artificial Intelligence – Progress in Artificial Intelligence (EPIA 2005)*, volume 3803 of *LNCIS*, pages 218–231, Covilha, Portugal, 2005. Springer.
  41. Raphael Volz. *Web Ontology Reasoning With Logic Databases*. PhD thesis, Universität Fridericiana zu Karlsruhe (TH), Germany, 2004.