

# Learning Semantic Web Rules: State of the Art and Directions of Research

Francesca A. Lisi and Floriana Esposito

Dipartimento di Informatica, Università degli Studi di Bari  
Via E. Orabona 4, 70125 Bari, Italy  
{lisi, esposito}@di.uniba.it

**Abstract.** The acquisition of Semantic Web rules is very demanding and can be automated though partially by applying Machine Learning (ML) algorithms. In this paper we provide a state-of-the-art survey of ML research relevant to this issue. In particular, we take a critical look at three ML frameworks that extend the methodological apparatus of Inductive Logic Programming to hybrid Knowledge Representation systems combining Description Logics and Clausal Logics. From the comparison of the three we draw general conclusions that suggest directions of research on this topic.

## 1 Motivation and Scope

Rules are currently in the focus within the Semantic Web architecture, and consequently interest and activity in this area has grown rapidly over recent years. They would allow the integration, transformation and derivation of data from numerous sources in a distributed, scalable, and transparent manner. The rules landscape features design aspects of rule markup; engineering of engines, translators, and other tools; standardization efforts, such as the recent Rules Interchange Format (RIF) activity at W3C; and applications. Rules complement and extend *ontologies* on the Semantic Web. They can be used in combination with ontologies, or as a means to specify ontologies. Rules are also frequently applied over ontologies, to draw inferences, express constraints, specify policies, react to events, discover new knowledge, transform data, etc. Rule markup languages enrich Web ontologies by supporting publishing rules on the Web, exchange rules between different systems and tools, share guidelines and policies, merge and maintain rulebases, and more. Yet, whereas the mark-up language OWL for Semantic Web ontologies is already undergoing the second round of the standardization process at W3C, the debate around a RIF is still ongoing. Because of the great variety in rule languages and rule engine technologies, this format will consist of a core language to be used along with a set of standard and non-standard extensions. These extensions need not all be combinable into a single unified language. As for the expressive power, two directions are followed: monotonic extensions towards full First Order Logic (FOL) and non-monotonic extensions based on the Logic Programming tradition, i.e. on *Clausal Logics*

(CLs). Since the design of OWL has been based on *Description Logics* (DLs) [1] (more precisely on the  $\mathcal{SH}$  family of very expressive DLs [13]), non-monotonic rule languages for the Semantic Web will most likely be inspired by *old* hybrid Knowledge Representation (KR) systems such as  $\mathcal{AL}$ -log [6] and CARIN [17] that integrate DLs and (fragments of) CLs. Such rule formalisms are of interest to us. Other uses of rules, e.g. in OWL 2, are beyond the scope of the paper.

The acquisition of Semantic Web rules is very demanding and can be automated though partially by applying Machine Learning (ML) algorithms. The ML approach known under the name of Inductive Logic Programming (ILP) [28] seems particularly promising for the following reasons. ILP is a form of Concept Learning rooted into Logic Programming. Thus it has been historically concerned with rule induction from examples and background knowledge within the KR framework of Horn Clausal Logic (HCL) and with the aim of prediction. The distinguishing feature of ILP, also with respect to other forms of Concept Learning, is the use of prior knowledge during the induction process. We claim that learning Semantic Web rules can be reformulated as learning rules by having ontologies as prior knowledge. This may motivate an interest of the Semantic Web community in ILP. In this paper we take a critical look at three ILP attempts at learning rules within hybrid DL-CL KR frameworks. From the comparative analysis of them we shall draw general conclusions that can be considered as guidelines for further ILP research of interest to the Semantic Web.

The paper is organized as follows. Section 2 first provides essential information on the ILP methodological apparatus for non informed readers. Section 3 briefly describes three major forms of integration of DLs and CLs. Section 4 provides a state-of-the-art survey of ILP proposals for the hybrid DL-CL formalisms considered in Section 3 and outlines directions of future work. Section 5 concludes the paper with final remarks. Appendixes A and B provide the basic notions of DLs and CLs, respectively.

## 2 Learning rules with ILP

Inductive Logic Programming (ILP) was born at the intersection between Concept Learning and Logic Programming. From Concept Learning it has inherited the inferential mechanisms for induction, the most prominent of which is *generalization*. A distinguishing feature of ILP with respect to other forms of Concept Learning is the use of background knowledge (BK). From Logic Programming it has borrowed the KR framework, i.e. HCL.

In Concept Learning, thus in ILP, generalization is traditionally viewed as search through a partially ordered space of inductive hypotheses [26]. According to this vision, an inductive hypothesis is a clausal theory and the induction of a single clause requires (i) structuring, (ii) searching and (iii) bounding the space of clauses [28]. First we focus on (i) by clarifying the notion of *ordering* for clauses. An ordering allows for determining which one, between two clauses, is more general than the other. Since partial orders are considered, uncomparable pairs of clauses are admitted. One such ordering is  *$\theta$ -subsumption* [29]: Given two

clauses  $C$  and  $D$ , we say that  $C$   $\theta$ -subsumes  $D$  if there exists a substitution  $\theta$ , such that  $C\theta \subseteq D$ <sup>1</sup>. Given the usefulness of BK, orders have been proposed that reckon with it. Among them, *generalized subsumption* [2] is of major interest to this paper: Given two definite clauses  $C$  and  $D$  standardized apart and a definite program  $\mathcal{K}$ , we say that  $C \succeq_{\mathcal{K}} D$  iff there exists a ground substitution  $\theta$  for  $C$  such that (i)  $head(C)\theta = head(D)\sigma$  and (ii)  $\mathcal{K} \cup body(D)\sigma \models body(C)\theta$  where  $\sigma$  is a Skolem substitution for  $D$  with respect to  $\{C\} \cup \mathcal{K}$ . Generalized subsumption is also called *semantic generality* in contrast to  $\theta$ -subsumption which is a purely syntactic generality. In the general case, generalized subsumption is undecidable and does not introduce a lattice on a set of clauses. Because of these problems,  $\theta$ -subsumption is more frequently used in ILP systems. Yet for DATALOG generalized subsumption is decidable and admits a least general generalization. Once structured, the space of hypotheses can be searched (ii) by means of refinement operators. A *refinement operator* is a function which computes a set of specializations or generalizations of a clause according to whether a top-down or a bottom-up search is performed. The two kinds of refinement operator have been therefore called *downward* and *upward*, respectively. The definition of refinement operators presupposes the investigation of the properties of the various orderings and is usually coupled with the specification of a declarative bias for bounding the space of clauses (iii). *Bias* concerns anything which constrains the search for theories, e.g. a *language bias* specifies syntactic constraints on the clauses in the search space.

Induction with ILP generalizes from individual instances/observations in the presence of BK, finding *valid hypotheses*. Validity depends on the underlying *setting*. At present, there exist several formalizations of induction in ILP that can be classified according to the following two orthogonal dimensions: the *scope of induction* (discrimination vs characterization) and the *representation of observations* (ground definite clauses vs ground unit clauses) [5]. *Discriminant induction* aims at inducing hypotheses with discriminant power as required in tasks such as classification. In classification, observations encompass both positive and negative examples. *Characteristic induction* is more suitable for finding regularities in a data set. This corresponds to learning from positive examples only. The second dimension affects the notion of *coverage*, i.e. the condition under which a hypothesis explains an observation. In *learning from entailment*, hypotheses are clausal theories, observations are ground definite clauses, and a hypothesis covers an observation if the hypothesis logically entails the observation. In *learning from interpretations*, hypotheses are clausal theories, observations are Herbrand interpretations (ground unit clauses) and a hypothesis covers an observation if the observation is a model for the hypothesis.

### 3 KR behind Semantic Web Rules

The definition of a rule language for the Semantic Web follows the tradition of KR research on *hybrid systems*, i.e. those systems which are constituted by

<sup>1</sup> See Appendix B for details of set notation for clauses.

two or more subsystems dealing with distinct portions of a single KB by performing specific reasoning procedures [10]. The motivation for investigating and developing such systems is to improve on two basic features of KR formalisms, namely *representational adequacy* and *deductive power*, by preserving the other crucial feature, i.e. *decidability*. Indeed DLs and CLs are FOL fragments incomparable as for the expressiveness and the semantics<sup>2</sup> but combinable under certain conditions [30]. In particular, combining DLs with HCL can easily yield to undecidability if the interaction scheme between the DL and the CL part of an hybrid KB does not fulfill the condition of *safeness*, i.e. does not solve the semantic mismatch between DLs and CLs [27,31].

A comprehensive study of the effects of combining DLs and CLs (more precisely, Horn rules) can be found in [17]. Special attention is devoted to the DL  $\mathcal{ALCN}\mathcal{R}$ . The results of the study can be summarized as follows: (i) answering conjunctive queries over  $\mathcal{ALCN}\mathcal{R}$  TBoxes is decidable, (ii) query answering in  $\mathcal{ALCN}\mathcal{R}$  extended with non-recursive DATALOG rules, where both concepts and roles can occur in rule bodies, is also decidable, as it can be reduced to answering a *union of conjunctive queries* (UCQ)<sup>3</sup>, (iii) if rules are recursive, query answering becomes undecidable, (iv) decidability can be regained by disallowing certain combinations of constructors in the logic, and (v) decidability can be regained by requiring rules to be *role-safe*, where at least one variable from each role literal must occur in some non-DL-atom. The integration framework proposed in [17] and known as **Carin** is therefore *unsafe*. Reasoning in CARIN is based on *constrained SLD-resolution*, i.e. an extension of SLD-resolution with a tableau calculus for DLs to deal with DL literals in the rules. Constrained SLD-refutation is a complete and sound method for answering *ground* queries.

$\mathcal{AL}\text{-log}$  [6] is a *safe* hybrid KR system that integrates  $\mathcal{ALC}$  [34] and DATALOG [4]. In particular, variables occurring in the body of rules may be constrained with  $\mathcal{ALC}$  concept assertions to be used as 'typing constraints'. This makes rules applicable only to explicitly named objects. As in CARIN, query answering is decided using the constrained SLD-resolution which however in  $\mathcal{AL}\text{-log}$  is decidable and runs in single non-deterministic exponential time.

The hybrid KR framework of  $\mathcal{DL}\text{+log}$  [32] allows for the tight integration of DATALOG<sup>- $\forall$</sup>  [7] with DLs. More precisely, it allows a DL KB to be extended with *weakly-safe* DATALOG<sup>- $\forall$</sup>  rules. The condition of weak safeness allows to overcome the main representational limits of the approaches based on the DL-safeness condition, e.g. the possibility of expressing UCQs, by keeping the integration scheme still decidable. To a certain extent,  $\mathcal{DL}\text{+log}$  is between  $\mathcal{AL}\text{-log}$  and CARIN. For  $\mathcal{DL}\text{+log}$  two semantics have been defined: a FOL semantics and a nonmonotonic (NM) semantics. In particular, the latter extends the stable model semantics

---

<sup>2</sup> Remind that the OWA holds for DLs whereas CWA is valid in CLs. Note that the OWA and CWA have a strong influence on the results of reasoning.

<sup>3</sup> A UCQ over a predicate alphabet  $P$  is a FOL sentence of the form  $\exists \mathbf{X}. conj_1(\mathbf{X}) \vee \dots \vee conj_n(\mathbf{X})$ , where  $\mathbf{X}$  is a tuple of variable symbols and each  $conj_i(\mathbf{X})$  is a set of atoms whose predicates are in  $P$  and whose arguments are either constants or variables from  $\mathbf{X}$ . A CQ corresponds to a UCQ in the case when  $n = 1$ .

**Table 1.** Three KR frameworks suitable for representing Semantic Web rules.

	Carin [17]	$\mathcal{AL}\text{-log}$ [6]	$\mathcal{DL}\text{+log}$ [32]
DL language	any DL	$\mathcal{ALC}$	any DL
CL language	Horn clauses	DATALOG clauses	DATALOG <sup><math>\neg\vee</math></sup> clauses
integration	unsafe	safe	weakly-safe
rule head literals	DL/Horn literals	DATALOG literal	DL/DATALOG literals
rule body literals	DL/Horn literals	$\mathcal{ALC}$ /DATALOG literals (no roles)	DL/DATALOG <sup><math>\neg</math></sup> literals
semantics reasoning	Herbrand models+DL models SLD-resolution+tableau calculus	idem idem	stable models+DL models stable model computation + Boolean CQ/UCQ containment
decidability	only for some instantiations	yes	for all instantiations with DLs for which the Boolean CQ/UCQ containment is decidable
implementation	n.a.(?)	n.a.(?)	n.a.

of DATALOG <sup>$\neg\vee$</sup>  [11]. According to it, DL-predicates are still interpreted under OWA, while DATALOG predicates are interpreted under CWA. Notice that, under both semantics, entailment can be reduced to satisfiability and, analogously, that CQ answering can be reduced to satisfiability. The problem statement of satisfiability for finite  $\mathcal{DL}\text{+log}$  KBs relies on the problem known as the *Boolean CQ/UCQ containment problem*<sup>4</sup> in DLs. It is shown that the decidability of reasoning in  $\mathcal{DL}\text{+log}$ , thus of ground query answering, depends on the decidability of the Boolean CQ/UCQ containment problem in  $\mathcal{DL}$ . Currently,  $\mathcal{SHIQ}\text{+log}$  is one of the most expressive decidable instantiations of  $\mathcal{DL}\text{+log}$  [12].

The distinguishing features of these three hybrid DL-CL KR frameworks are summarized in Table 1.

## 4 ILP for Learning Semantic Web Rules

Hybrid KR systems combining DLs and (fragments of) HCL have very recently attracted some attention in the ILP community.

### 4.1 State of the Art

Only three ILP frameworks have been proposed that adopt a hybrid DL-CL representation for both hypotheses and background knowledge: [33] chooses CARIN- $\mathcal{ALN}$ , [18] resorts to  $\mathcal{AL}\text{-log}$ , and [20] builds upon  $\mathcal{SHIQ}\text{+log}$ . A comparative analysis of the three is reported in Table 2. They can be considered as attempts at accommodating ontologies in ILP. Indeed, they can deal with  $\mathcal{ALN}$ ,  $\mathcal{ALC}$ <sup>5</sup>, and  $\mathcal{SHIQ}$  ontologies respectively. Learning Semantic Web rules with ILP can be reformulated as learning rules by having ontologies as prior knowledge. Both proposals have been guided by a similar consideration when extending previous work in ILP to hybrid DL-CL KR frameworks.

<sup>4</sup> This problem was called *existential entailment* in [17].

<sup>5</sup> We remind the reader that  $\mathcal{ALN}$  and  $\mathcal{ALC}$  are incomparable DLs.

**Learning in Carin- $\mathcal{ALN}$**  The framework proposed in [33] focuses on discriminant induction and adopts the ILP setting of learning from interpretations. Hypotheses are represented as CARIN- $\mathcal{ALN}$  non-recursive rules with a Horn literal in the head that plays the role of target concept. The coverage relation of hypotheses against examples adapts the usual one in learning from interpretations to the case of hybrid CARIN- $\mathcal{ALN}$  BK. The generality relation between two hypotheses is defined as an extension of generalized subsumption. Procedures for testing both the coverage relation and the generality relation are based on the existential entailment algorithm of CARIN. Following [33], Kietz studies the learnability of CARIN- $\mathcal{ALN}$ , thus providing a pre-processing method which enables ILP systems to learn CARIN- $\mathcal{ALN}$  rules [16].

**Learning in  $\mathcal{AL}$ -log** In [18], hypotheses are represented as constrained DATALOG clauses that are linked, connected (or range-restricted), and compliant with the bias of Object Identity (OI)<sup>6</sup>. As opposite to [33], this framework is general, meaning that it is valid whatever the scope of induction (description/prediction) is. Therefore the literal in the head of hypotheses represents a concept to be either discriminated from others (*discriminant induction*) or characterized (*characteristic induction*). The generality relation for one such hypothesis language is an adaptation of generalized subsumption, named  $\mathcal{B}$ -subsumption, to the  $\mathcal{AL}$ -log KR framework. It gives raise to a quasi-order and can be checked with a decidable procedure based on constrained SLD-resolution [22]. Coverage relations for both ILP settings of learning from interpretations and learning from entailment have been defined on the basis of query answering in  $\mathcal{AL}$ -log [19]. As opposite to [33], the framework has been implemented in an ILP system [24]. More precisely, an instantiation of it for the case of *characteristic induction from interpretations* has been considered. Indeed, the system supports a variant of a very popular data mining task - frequent pattern discovery - where rich prior conceptual knowledge is taken into account during the discovery process in order to find patterns at multiple levels of description granularity. The search through the space of patterns represented as unary conjunctive queries in  $\mathcal{AL}$ -log and organized according to  $\mathcal{B}$ -subsumption is performed by applying an ideal downward refinement operator [23].

**Learning in  $SHIQ$ +log** The ILP framework presented in [20] represents hypotheses as  $SHIQ$ +log rules restricted to positive DATALOG and organizes them according to a generality ordering inspired by generalized subsumption. The resulting hypothesis space can be searched by means of refinement operators either top-down or bottom-up. Analogously to [18], this framework encompasses both scopes of induction but, differently from [18], it assumes the ILP setting of learning from interpretations only. Both the coverage relation and the generality

---

<sup>6</sup> The OI bias can be considered as an extension of the UNA from the semantic level to the syntactic one of  $\mathcal{AL}$ -log. It can be the starting point for the definition of either an equational theory or a quasi-order for constrained DATALOG clauses.

relation boil down to query answering in  $\mathcal{SHIQ}+\log$ , thus can be reformulated as satisfiability problems. Compared to [33] and [18], this framework shows an added value which can be summarized as follows. First, it relies on a more expressive DL, i.e.  $\mathcal{SHIQ}$ . Second, it allows for inducing definitions for new DL concepts, i.e. rules with a  $\mathcal{SHIQ}$  literal in the head. Third, it adopts a tighter form of integration between the DL and the CL part, i.e. the weakly-safe one.

**Table 2.** Three ILP frameworks suitable for learning Semantic Web rules.

	Learning in Carin- $\mathcal{ALN}$ [33]	Learning in $\mathcal{AL}\text{-log}$ [18]	Learning in $\mathcal{SHIQ}+\log$ [20]
prior knowledge	CARIN- $\mathcal{ALN}$ KB	$\mathcal{AL}\text{-log}$ KB	$\mathcal{SHIQ}+\log$ KB
ontology lang.	$\mathcal{ALN}$	$\mathcal{ALC}$	$\mathcal{SHIQ}$
rule lang.	Horn clauses	DATALOG clauses	DATALOG clauses
hypothesis lang.	CARIN- $\mathcal{ALN}$ non-recursive rules	constrained DATALOG clauses	$\mathcal{SHIQ}+\log$ non-recursive rules
target predicate	Horn literal	DATALOG literal	$\mathcal{SHIQ}$ /DATALOG literal
setting	interpretations	interpretations/entailment	interpretations
induction	predictive	predictive/descriptive	predictive/descriptive
generality order	extension of [2] to CARIN- $\mathcal{ALN}$	extension of [2] to $\mathcal{AL}\text{-log}$	extension of [2] to $\mathcal{SHIQ}+\log$
coverage test	CARIN- $\mathcal{ALN}$ query answering	$\mathcal{AL}\text{-log}$ query answering	$\mathcal{SHIQ}+\log$ query answering
ref. operators	n.a.	downward	downward/upward
implementation	n.a.	partial	n.a.
application	n.a.	yes	n.a.

## 4.2 Directions of Research

From the comparative analysis of the ILP frameworks reviewed in Section 4.1, a common feature emerges: All proposals resort to Buntine’s generalized subsumption and extend it in a non-trivial way. This choice is due to the fact that, among the semantic generality orders in ILP, generalized subsumption applies only to definite clauses, therefore suits well the hypothesis language in all three frameworks. Following these guidelines, new ILP frameworks can be designed to deal with more or differently expressive hybrid DL-CL languages according to the DL chosen (e.g., learning CARIN- $\mathcal{ALCN}\mathcal{R}$  rules), or the clausal language chosen (e.g., learning recursive CARIN rules), or the integration scheme (e.g., learning CARIN rules with DL-literals in the head). An important requirement will be the definition of a *semantic* generality relation for hypotheses to take into account the background knowledge. Of course, generalized subsumption may turn out to be not suitable for these upcoming ILP frameworks, e.g. for the case of learning disjunctive  $\mathcal{DL}+\log$  rules. Speaking of which, the inclusion of nonmonotonic features like negation and disjunction - admissible in  $\mathcal{DL}+\log$  *full* - into the language for hypotheses and background knowledge will strengthen the ability of the ILP frameworks to deal with incomplete knowledge by performing an inductive form of commonsense reasoning. One such ability can turn out to be useful in the Semantic Web, and complementary to reasoning with uncertainty and under inconsistency.

Also it would be interesting to investigate how the nature of rules (i.e., the intended context of usage) may impact the learning process as for the scope

of induction and other variables in the learning problem statement. E.g., the problem of learning  $\mathcal{AL}$ -log rules for classification purposes differ greatly from the apparently similar learning problem faced in [24].

Besides theoretical issues, most future work will have to be devoted to implementation and application. When moving to practice, issues like efficiency and scalability become of paramount importance. These concerns may drive the attention of ILP research towards less expressive hybrid KR frameworks in order to gain in tractability, e.g. instantiations of  $\mathcal{DL}$ +log with DL-Lite [3]. Applications can come out of some of the many use cases for Semantic Web rules specified by the RIF W3C Working Group. Considering the current trend to have rules within ontologies rather than on top of ontologies, it is worthwhile to explore the possibility of learning rules for ontology evolution according to the proof-of-concept scenario described in [21].

## 5 Final remarks

Building rules on top of ontologies for the Semantic Web poses several challenges not only to KR researchers investigating suitable hybrid DL-CL formalisms but also to the ML community which has been historically interested in application areas where the Knowledge Acquisition bottleneck is particularly severe.

In this paper, we have revised the ML literature addressing the problem of learning hybrid DL-CL rules. Only three ILP works have been found that propose a solution to this problem [33,18,20]. They adopt CARIN- $\mathcal{ALN}$ ,  $\mathcal{AL}$ -log and  $SHIQ$ +log as KR framework, respectively. Note that matching Table 2 against Table 1 one may figure out what is the state-of-the-art and what are the directions of research on Semantic Web rules from the ML viewpoint. Also he/she can get suggestions on what is the most appropriate among these ILP frameworks to be implemented for a certain intended application.

Closely related to DL-CL KR systems are the hybrid formalisms arising from the study of many-sorted logics, where a FOL language is combined with a sort language which can be regarded as an elementary DL [8]. In this respect the study of a sorted downward refinement [9] can be also considered as a contribution to the problem of interest to this paper. Finally, some work has been done on discovering frequent association patterns [15] in the form of DL-safe rules [27].

## References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications (2nd ed.)*. Cambridge University Press, 2007.
2. W. Buntine. Generalized subsumption and its application to induction and redundancy. *Artificial Intelligence*, 36(2):149–176, 1988.
3. D. Calvanese, M. Lenzerini, R. Rosati, and G. Vetere. DL-Lite: Practical Reasoning for Rich DLs. In V. Haarslev and R. Möller, editors, *Proceedings of the 2004 International Workshop on Description Logics (DL2004)*, volume 104 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.



4. S. Ceri, G. Gottlob, and L. Tanca. *Logic Programming and Databases*. Springer, 1990.
5. L. De Raedt and L. Dehaspe. Clausal Discovery. *Machine Learning*, 26(2–3):99–146, 1997.
6. F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf.  $\mathcal{AL}$ -log: Integrating Datalog and Description Logics. *Journal of Intelligent Information Systems*, 10(3):227–252, 1998.
7. T. Eiter, G. Gottlob, and H. Mannila. Disjunctive DATALOG. *ACM Transactions on Database Systems*, 22(3):364–418, 1997.
8. A.M. Frisch. The substitutional framework for sorted deduction: Fundamental results on hybrid reasoning. *Artificial Intelligence*, 49:161–198, 1991.
9. A.M. Frisch. Sorted downward refinement: Building background knowledge into a refinement operator for inductive logic programming. In S. Džeroski and P. Flach, editors, *Inductive Logic Programming*, volume 1634 of *Lecture Notes in Artificial Intelligence*, pages 104–115. Springer, 1999.
10. A.M. Frisch and A.G. Cohn. Thoughts and afterthoughts on the 1988 workshop on principles of hybrid reasoning. *AI Magazine*, 11(5):84–87, 1991.
11. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–386, 1991.
12. B. Glimm, I. Horrocks, C. Lutz, and U. Sattler. Conjunctive query answering for the description logic  $\mathcal{SHIQ}$ . *Journal of Artificial Intelligence Research*, 31:151–198, 2008.
13. I. Horrocks, P.F. Patel-Schneider, and F. van Harmelen. From  $\mathcal{SHIQ}$  and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.
14. I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.
15. J. Józefowska, A. Lawrynowicz, and T. Lukaszewski. Towards discovery of frequent patterns in description logics with rules. In A. Adi, S. Stoutenburg, and S. Tabet, editors, *Rules and Rule Markup Languages for the Semantic Web*, volume 3791 of *Lecture Notes in Computer Science*, pages 84–97. Springer, 2005.
16. J.-U. Kietz. Learnability of description logic programs. In S. Matwin and C. Sammut, editors, *Inductive Logic Programming*, volume 2583 of *Lecture Notes in Artificial Intelligence*, pages 117–132. Springer, 2003.
17. A.Y. Levy and M.-C. Rousset. Combining Horn rules and description logics in CARIN. *Artificial Intelligence*, 104:165–209, 1998.
18. F.A. Lisi. Building Rules on Top of Ontologies for the Semantic Web with Inductive Logic Programming. *Theory and Practice of Logic Programming*, 8(03):271–300, 2008.
19. F.A. Lisi and F. Esposito. Efficient Evaluation of Candidate Hypotheses in  $\mathcal{AL}$ -log. In R. Camacho, R. King, and A. Srinivasan, editors, *Inductive Logic Programming*, volume 3194 of *Lecture Notes in Artificial Intelligence*, pages 216–233. Springer, 2004.
20. F.A. Lisi and F. Esposito. Foundations of Onto-Relational Learning. In F. Železný and N. Lavrač, editors, *Inductive Logic Programming*, volume 5194 of *Lecture Notes in Artificial Intelligence*, pages 158–175. Springer, 2008.
21. F.A. Lisi and F. Esposito. Learning  $\mathcal{SHIQ}$ +log Rules for Ontology Evolution. In A. Gangemi, J. Keizer, V. Presutti, and H. Stoermer, editors, *Semantic Web Applications and Perspectives (SWAP2008)*, volume 426 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

22. F.A. Lisi and D. Malerba. Bridging the Gap between Horn Clausal Logic and Description Logics in Inductive Learning. In A. Cappelli and F. Turini, editors, *AI\*IA 2003: Advances in Artificial Intelligence*, volume 2829 of *Lecture Notes in Artificial Intelligence*, pages 49–60. Springer, 2003.
23. F.A. Lisi and D. Malerba. Ideal Refinement of Descriptions in  $\mathcal{AL}$ -log. In T. Horvath and A. Yamamoto, editors, *Inductive Logic Programming*, volume 2835 of *Lecture Notes in Artificial Intelligence*, pages 215–232. Springer, 2003.
24. F.A. Lisi and D. Malerba. Inducing Multi-Level Association Rules from Multiple Relations. *Machine Learning*, 55:175–210, 2004.
25. J.W. Lloyd. *Foundations of Logic Programming*. Springer, 2nd edition, 1987.
26. T.M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.
27. B. Motik, U. Sattler, and R. Studer. Query Answering for OWL-DL with Rules. In S.A. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *The Semantic Web*, volume 3298 of *Lecture Notes in Computer Science*, pages 549–563. Springer, 2004.
28. S.-H. Nienhuys-Cheng and R. de Wolf. *Foundations of Inductive Logic Programming*, volume 1228 of *Lecture Notes in Artificial Intelligence*. Springer, 1997.
29. G.D. Plotkin. A note on inductive generalization. *Machine Intelligence*, 5:153–163, 1970.
30. R. Rosati. On the decidability and complexity of integrating ontologies and rules. *Journal of Web Semantics*, 3(1):61–73, 2005.
31. R. Rosati. Semantic and computational advantages of the safe integration of ontologies and rules. In F. Fages and S. Soliman, editors, *Principles and Practice of Semantic Web Reasoning*, volume 3703 of *Lecture Notes in Computer Science*, pages 50–64. Springer, 2005.
32. R. Rosati.  $\mathcal{DL}+\text{log}$ : Tight Integration of Description Logics and Disjunctive Datalog. In P. Doherty, J. Mylopoulos, and C.A. Welty, editors, *Proc. of Tenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 68–78. AAAI Press, 2006.
33. C. Rouveirol and V. Ventos. Towards Learning in CARIN- $\mathcal{ALN}$ . In J. Cussens and A. Frisch, editors, *Inductive Logic Programming*, volume 1866 of *Lecture Notes in Artificial Intelligence*, pages 191–208. Springer, 2000.
34. M. Schmidt-Schauss and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

## A Description Logics

DLs are a family of decidable FOL fragments that allow for the specification of knowledge in terms of classes (*concepts*), binary relations between classes (*roles*), and instances (*individuals*) [1]. Complex concepts can be defined from atomic concepts and roles by means of constructors (see Table 3). E.g., concept descriptions in the basic DL  $\mathcal{AL}$  are formed according to only the constructors of atomic negation, concept conjunction, value restriction, and limited existential restriction. The DLs  $\mathcal{ALC}$  and  $\mathcal{ALN}$  are members of the  $\mathcal{AL}$  family [34]. The former extends  $\mathcal{AL}$  with (arbitrary) concept negation (also called complement and equivalent to having both concept union and full existential restriction), whereas the latter with number restriction. The DL  $\mathcal{ALCN}$  adds to the constructors inherited from  $\mathcal{ALC}$  and  $\mathcal{ALN}$  a further one: role intersection (see

**Table 3.** Syntax and semantics of DLs.

bottom (resp. top) concept	$\perp$ (resp. $\top$ )	$\emptyset$ (resp. $\Delta^{\mathcal{I}}$ )
atomic concept	$A$	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
(abstract) role	$R$	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
(abstract) inverse role	$R^{-}$	$(R^{\mathcal{I}})^{-}$
(abstract) individual	$a$	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
concept negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
concept intersection	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
concept union	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
value restriction	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y (x, y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
existential restriction	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y (x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
at least number restriction	$\geq nR$	$\{x \in \Delta^{\mathcal{I}} \mid  \{y \mid (x, y) \in R^{\mathcal{I}}\}  \geq n\}$
at most number restriction	$\leq nR$	$\{x \in \Delta^{\mathcal{I}} \mid  \{y \mid (x, y) \in R^{\mathcal{I}}\}  \leq n\}$
at least qualif. number restriction	$\geq nS.C$	$\{x \in \Delta^{\mathcal{I}} \mid  \{y \in C^{\mathcal{I}} \mid (x, y) \in S^{\mathcal{I}}\}  \geq n\}$
at most qualif. number restriction	$\leq nS.C$	$\{x \in \Delta^{\mathcal{I}} \mid  \{y \in C^{\mathcal{I}} \mid (x, y) \in S^{\mathcal{I}}\}  \leq n\}$
role intersection	$R_1 \sqcap R_2$	$R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}}$
concept equivalence axiom	$C_1 \equiv C_2$	$C_1^{\mathcal{I}} = C_2^{\mathcal{I}}$
concept subsumption axiom	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
role equivalence axiom	$R \equiv S$	$R^{\mathcal{I}} = S^{\mathcal{I}}$
role inclusion axiom	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
concept assertion	$a : C$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$\langle a, b \rangle : R$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
individual equality assertion	$a \approx b$	$a^{\mathcal{I}} = b^{\mathcal{I}}$
individual inequality assertion	$a \not\approx b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$

Table 3). Conversely, in the DL *SHIQ* [14] it is allowed to invert roles and to express qualified number restrictions of the form  $\geq nS.C$  and  $\leq nS.C$  where  $S$  is a simple role (see Table 3). Also transitivity holds for roles.

A DL knowledge base (KB) can state both is-a relations between concepts (*axioms*) and instance-of relations between individuals (resp. couples of individuals) and concepts (resp. roles) (*assertions*). Concepts and axioms form the so-called TBox (Terminological Box, or intensional part of a DL KB) whereas individuals and assertions form the so-called ABox (Assertional Box, or extensional part of a DL KB). A *SHIQ* KB encompasses also a RBox (Role Box) which consists of axioms concerning abstract roles. The direct semantics of DLs is shown in Table 3. An *interpretation*  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  for a DL KB consists of a domain  $\Delta^{\mathcal{I}}$  and a mapping function  $\cdot^{\mathcal{I}}$ . Individuals are mapped to elements of  $\Delta^{\mathcal{I}}$  such that  $a^{\mathcal{I}} \neq b^{\mathcal{I}}$  if  $a \neq b$  (*Unique Names Assumption* (UNA)). Yet in *SHIQ* UNA does not hold by default [12]. Thus individual equality (inequality) assertions may appear in a *SHIQ* KB (see Table 3). Also the KB represents many different interpretations, i.e. all its models. This is coherent with the *Open*

*World Assumption* (OWA) that holds in FOL semantics. The main reasoning task for a DL KB is the *consistency check* that is performed by applying decision procedures based on tableau calculus. Decidability of reasoning is crucial in DLs.

## B Clausal Logics

The basic element in Clausal Logics (CLs) is the *atom* of the form  $p(t_1, \dots, t_{k_i})$  such that each  $p$  is a predicate symbol and each  $t_j$  is a term. A *term* is either a constant or a variable or a more complex term obtained by applying a functor to simpler term. Constant, variable, functor and predicate symbols belong to mutually disjoint alphabets. A *literal* is an atom either negated or not. A *clause* is a universally quantified disjunction of literals. Usually the universal quantifiers are omitted to simplify notation, therefore a clause is a disjunctive formula like

$$h_1 \vee \dots \vee h_n \vee \neg b_1 \vee \dots \vee \neg b_m$$

or a set of literals

$$\{h_1, \dots, h_n, \neg b_1, \dots, \neg b_m\}$$

or, according to the most widely used alternative notation, an implication

$$h_1 \vee \dots \vee h_n \leftarrow b_1 \wedge \dots \wedge b_m$$

whose right-hand side and left-hand side are called *head* and *body* of the clause, respectively. A *program* is a set of clauses. The most studied CL is called Horn Clausal Logic (HCL). It admits only so-called *definite clauses*, i.e. clauses with  $n = 1$ , which are called *rules* and *facts* for  $m > 0$  and  $m = 0$  respectively. Definite clauses are at the basis of logic programming [25] and deductive databases [4].

In the logic programming context, the model-theoretic semantics of HCL is based on the notion of Herbrand interpretation. The corresponding proof-theoretic semantics is based on the *Closed World Assumption* (CWA) and is the starting point for a deductive reasoning mechanism, i.e. *SLD-resolution*, which is sound and complete by refutation.

In the deductive database context, a version of HCL free from functors and recursion leads to DATALOG [4]. The main reasoning task in DATALOG is query answering. A *query*  $Q$  to a DATALOG program  $D$  is a DATALOG clause with  $n = 0$  and  $m > 0$ . An *answer* to a query  $Q$  is a substitution  $\theta$  for the variables of  $Q$ . An answer is correct with respect to the DATALOG program  $D$  if  $D \models Q\theta$ . The *answer set* to a query  $Q$  is the set of answers to  $Q$  that are correct w.r.t.  $D$  and such that  $Q\theta$  is ground. Answers are computed by SLD-refutation. Disjunctive DATALOG (DATALOG<sup>∨</sup>) is a variant of DATALOG where disjunctions may appear in the rule heads [7]. Advanced versions (DATALOG<sup>∨,¬</sup>) also allow for negated literals in the rule bodies, which can be handled according to a semantics for negation in CL, e.g. *stable model semantics* [11]. Under such semantics, a DATALOG<sup>∨,¬</sup> program may have several alternative models (but possibly none), each corresponding to a possible view of the reality.