Higher-order Description Logics for Learning and Mining in Complex Domains*

Francesca A. Lisi

Dipartimento di Informatica &
Centro Interdipartimentale di Logica e Applicazioni (CILA)
Università degli Studi di Bari "Aldo Moro", Italy
FrancescaAlessandra.Lisi@uniba.it

Abstract. This short paper summarizes the work I have done over the last years on the use of higher-order Description Logics (DLs) for learning and mining in complex domains. In particular, the work proposes higher-order DLs as a means for metamodeling and metaquerying in Concept Learning and Knowledge Graph Mining, respectively.

Keywords: Higher-order Description Logics · Concept Learning · Knowledge Graph Mining.

1 Introduction

Most learning and mining problems can be reformulated as Constraint Satisfaction Problems (CSPs) or Optimization Problems (OPs). So, problem solving in this context could in principle take advantage of generic solvers, by exclusively using a description of the relevant domain knowledge and the conditions imposed by the problem to be solved. However, in spite of focusing on problem specification, research in this area has traditionally focused on designing effective specific algorithms for solving the problem in hand. As stressed by De Raedt [7], there is an increasing interest in providing the user with languages for learning and mining. This change of perspective claims for a *model+solver* approach to learning and mining problems, in which the user specifies the problem by means of a *declarative modeling language* and the system automatically transforms such models into a format that can be used by a *solver* to efficiently generate a solution. For instance, constraint programming has been successfully applied to itemset mining problems (see, *e.g.*, [12] for a comprehensive account). Another notable example is the framework of Meta-Interpretive Learning (MIL) [26]. MIL uses descriptions in the form of meta-rules (expressed in a higher-order dyadic Datalog fragment) with procedural constraints incorporated within a meta-interpreter, which could be eventually implemented by relying on *Answer Set Programming* (ASP) solvers (see [10] for an updated overview).

This short paper summarizes the work I have done over the last years on the use of higher-order Description Logics (DLs) for learning and mining in complex domains. In particular, the work proposes higher-order DLs as a means for metamodeling and metaquerying in Concept Learning and Knowledge Graph Mining, respectively.

2 Learning and Mining in Complex Domains

Machine Learning (ML) and Data Mining (DM) algorithms both look for regularities in data, by means of some *inductive reasoning* mechanism such as *generalization*. However, it is conventional to distinguish between the two classes of algorithms as for the scope of induction. In particular, learning algorithms usually aim at *prediction* on unseen data, whereas mining algorithms have typically the scope of mere *description* of the given data.

Structure is inherent to data and knowledge in complex domains, and needs appropriate means for representation. Among the many formalisms used for representing structured knowledge, one of the most popular is the family of *Description Logics* (DLs) [1], which has been the starting point for the definition of the ontology language OWL.[1] A DL *knowledge base* (or equivalently, an OWL ontology) is a collection of logical axioms and assertions. RDF[2] is another popular formalism for structured knowledge, which however is less expressive than OWL. A *knowledge graph* (KG) is a huge collection of RDF triples. KGs can be interlinked and overall they implement the so-called Web of Data, *i.e.*, the vision of the World Wide Web (WWW) as a distributed database system.

Structured knowledge poses several challenges to learning and mining algorithms. In the following subsections I will briefly introduce the two cases of interest for this work, namely Concept Learning and Knowledge Graph Mining.

2.1 Concept Learning

Concept Learning deals with inferring the general definition of a category based on members (positive examples) and nonmembers (negative examples) of this category. In Concept Learning, the key inferential mechanism for induction is *generalization as search* through a partially ordered space of inductive hypotheses [23]. A popular form of Concept Learning is the one known under the name of *Inductive Logic Programming* (ILP) [25] where the hypotheses are typically expressed in the form of first-order Horn clauses (or other fragments of first-order logic). A distinguishing feature of ILP with respect to other forms of Concept Learning is the use of prior knowledge of the domain of interest, called *background knowledge* (BK), during the search for hypotheses. In ILP it is also common practice to exploit some *declarative bias* to, *e.g.*, constrain the language of hypotheses.

[1] https://www.w3.org/TR/owl2-overview/
[2] https://www.w3.org/RDF/

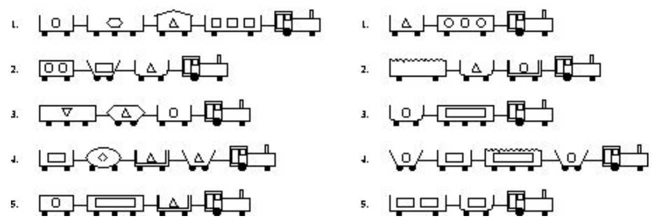

Fig. 1. Michalski's example of eastbound (left) and westbound (right) trains (illustration taken from [22]).

Concept Learning in DLs has been paid increasing attention since the 90s. Early work essentially focused on demonstrating the PAC-learnability for various terminological languages derived from the CLASSIC DL (see, *e.g.*, [3]). Later works such as [2,14] have followed the *generalization as search* approach by extending the methodological apparatus of ILP to DL languages. More recently there has been a renewed interest in more theoretical work (see, *e.g.*, [13]).

There are several variants of the Concept Learning problem in the DL context. The variant I consider as a showcase in this paper is the supervised one. In the following, the set of all individuals occurring in $\mathcal{A}$ and the set of all individuals occurring in $\mathcal{A}$ that are instances of a given concept C w.r.t. $\mathcal{K}$ are denoted by $\mathsf{Ind}(\mathcal{A})$ and $\mathsf{Retr}_{\mathcal{K}}(C)$, respectively.

Definition 1 (Concept Induction - CSP version). *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a DL KB. Given a (new) target concept name C, a set of positive and negative examples $\mathsf{Ind}^+_C(\mathcal{A}) \cup \mathsf{Ind}^-_C(\mathcal{A})$, and a concept description language $\mathcal{DL}_{\mathcal{H}}$, the CSP version of the* Concept Induction *problem (denoted by CI-CSP) is to find a concept definition $C \equiv D$ with $D \in \mathcal{DL}_{\mathcal{H}}$ such that: (i) $\mathcal{K} \models (a : D) \quad \forall a \in \mathsf{Ind}^+_C(\mathcal{A})$, and (ii) $\mathcal{K} \models (b : \neg D) \quad \forall b \in \mathsf{Ind}^-_C(\mathcal{A})$.*

Example 1. For illustrative purposes of the CI-CSP problem, let us consider a very popular classification problem proposed 40 years ago by Ryszard Michalski [22] and illustrated in Figure 1. Here, 10 trains are described, out of which 5 are eastbound and 5 are westbound. The aim of this problem is to find the discriminating features between these two classes referred to as `EastTrain` and `WestTrain` (or, more briefly, as `ET` and `WT`) from now on.

For the purpose of this case study, let us consider an $\mathcal{ALCO}$ ontology, *trains2*, encoding the original Trains data set. [3] With reference to *trains2* (which therefore will play the role of $\mathcal{K}$ as in Def. 1), we might want to induce a $\mathcal{SROIQ}$ concept definition for the target concept name `ET` (*i.e.*, the language of hypotheses is some $\mathcal{SROIQ}_{\mathcal{H}}$ based on $\mathcal{SROIQ}$) from the following positive and negative examples:

[3] `http://archive.ics.uci.edu/ml/datasets/Trains`

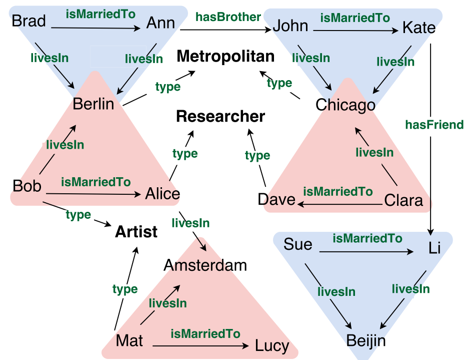

Fig. 2. Fragment of a knowledge graph (taken from [30]).

– $\mathsf{Ind}^{+}_{\mathtt{ET}}(\mathcal{A}) = \{\mathtt{et1}, \ldots, \mathtt{et5}\} \subseteq \mathsf{Retr}_{\mathcal{K}}(\mathtt{ET})$
– $\mathsf{Ind}^{-}_{\mathtt{ET}}(\mathcal{A}) = \{\mathtt{wt1}, \ldots, \mathtt{wt5}\} \subseteq \mathsf{Retr}_{\mathcal{K}}(\neg\mathtt{ET})$

Note that the 5 positive examples for `ET` are negative examples for `WT` and vice versa.

2.2 Knowledge Graph Mining

The analysis of data contained in a KG (referred to as *KG Mining*) is preliminary to several crucial maintenance tasks, notably the automated completion of the graph (aka *link prediction*), which pose several challenges due to the open and distributed environment of the WWW infrastructure. In the KG community approaches for link prediction are divided into statistics-based (see [27] for an overview), and logic-based (e.g., [9,30]). The latter, which are the closest to the work reported in this paper, basically extend and adapt previous work in ILP on relational association rule mining. However, they differ in the expressiveness of the mined rules. AMIE+ [9] can mine only Horn rules, whereas the methodology presented in [30] can address the case of nonmonotonic rules.

Example 2. In the context of link prediction, the following rule

$$isMarriedTo(X, Y), livesIn(X, Z) \Rightarrow livesIn(Y, Z) \quad (1)$$

can be mined from the KG in Fig. 2 and applied to derive new facts such as $livesIn(alice, berlin)$, $livesIn(dave, chicago)$ and $livesIn(lucy, amsterdam)$ to be used for completing the graph.

3 Higher-order DLs for Learning and Mining

In several applications there is a need for modeling and reasoning about meta-concepts, *i.e.*, concepts whose instances are themselves concepts, and meta-properties, *i.e.*, relationships between meta-concepts. *Metamodeling* addresses this need. Indeed, it allows one to treat concepts and properties as first-order citizens, and to see them as individuals whose properties can be asserted and reasoned upon. A common feature to metamodeling approaches is the use of logical languages with higher-order constructs for a correct representation of concepts and properties at the meta-level. *Metaquerying* is a special case of domain metamodeling. This is the case where the knowledge base does not contain any axiom regarding meta-concepts or meta-properties, but the query language allows for using meta-concepts and meta-properties, so that concepts and properties in the knowledge base can match the variables in the query, and may thus be returned as answers to the query. This mechanism allows to express queries that are beyond first-order logic.

Metamodeling (and metaquerying) has recently attracted an increasing interest in the Knowledge Representation (KR) community, thus giving rise to a stream of research aimed at extending DLs with higher-order features (see, *e.g.*, [28,24,4,5,15]). In particular, Colucci *et al.* [4] introduce second-order features in DLs under the Henkin semantics for modeling several forms of non-standard reasoning. The Henkin style shows a desirable feature, *i.e.*, the expressive power of the language actually remains first-order.

In the following two subsections I briefly report the main achievements of my research on metamodeling and metaquerying by means of higher-order DLs in the context of Concept Learning and Knowledge Graph Mining.

3.1 Metamodeling in Concept Learning

In [16], I have extended Colucci *et al.*'s work on non-standard reasoning in DLs [4] to several variants of Concept Learning, thus being the first to propose higher-order DLs under Henkin semantics as a means for metamodeling in ML. The idea is that each of these variants, besides being considered as non-standard reasoning tasks, can be reformulated as a CSP or even as an OP. For the sake of illustration I will focus on the case of $\mathsf{CI\text{-}CSP}$.

Following Def. 1, let us assume that $\mathsf{Ind}^{+}_{C}(\mathcal{A}) = \{a_1, \ldots, a_m\}$ and $\mathsf{Ind}^{-}_{C}(\mathcal{A}) = \{b_1, \ldots, b_n\}$. A concept $D \in \mathcal{DL}_{\mathcal{H}}$ is a correct concept definition for the target concept name C w.r.t. $\mathsf{Ind}^{+}_{C}(\mathcal{A})$ and $\mathsf{Ind}^{-}_{C}(\mathcal{A})$ iff it is a solution for the following second-order concept expression:

$$\gamma_{\mathsf{CI\text{-}CSP}} := (a_1 : X) \wedge \ldots \wedge (a_m : X) \wedge (b_1 : \neg X) \wedge \ldots \wedge (b_n : \neg X) \tag{2}$$

that is, iff D can be a valid assignment for the concept variable X. The $\mathsf{CI\text{-}CSP}$ problem can be modeled with the following second-order formula

$$\phi_{\mathsf{CI\text{-}CSP}} := \exists X.\gamma_{\mathsf{CI\text{-}CSP}} \tag{3}$$

The solvability of a $\mathsf{CI\text{-}CSP}$ problem is therefore based on the satisfiability of the second-order formula being used for modeling the problem.

In [19,20], the proposed *model+solver* approach combines the efficacy of higher-order DLs in metamodeling (as shown in [16]) with the efficiency of ASP solvers in dealing with CSPs and OPs. The encoding into ASP is possible under the *fixed-domain semantics* [8], a non-standard model-theoretic semantics for DLs which has been proposed in order to correctly address CSPs in OWL.

Example 3. According to (2), the intended CI-CSP problem of Example 1 corresponds to the following second-order concept expression:

$$\gamma^{\mathsf{ET}}_{\mathsf{CI\text{-}CSP}} := (\mathtt{et1} : X) \wedge \ldots \wedge (\mathtt{et5} : X) \wedge (\mathtt{wt1} : \neg X) \wedge \ldots \wedge (\mathtt{wt5} : \neg X) \quad (4)$$

The problem is then solvable if the following second-order formula:

$$\phi^{\mathsf{ET}}_{\mathsf{CI\text{-}CSP}} := \exists X.\gamma^{\mathsf{ET}}_{\mathsf{CI\text{-}CSP}} \quad (5)$$

is true in $\mathcal{SROIQ}_{\mathcal{H}}$, *i.e.*, if there exists a solution to $\gamma^{\mathsf{ET}}_{\mathsf{CI\text{-}CSP}}$ in $\mathcal{SROIQ}_{\mathcal{H}}$.

Let us now assume that $\mathcal{SROIQ}_{\mathcal{H}}$ is the set of all $\mathcal{SROIQ}$ concept expressions that can be generated starting from the atomic concept and role names occurring in *trains2* (except, of course, for the target concept name). Among the concepts belonging to $\mathcal{SROIQ}_{\mathcal{H}}$ and satisfying $\gamma^{\mathsf{ET}}_{\mathsf{CI\text{-}CSP}}$, there is

$$\exists\, \mathtt{hasCar}.(\mathtt{ClosedCar} \sqcap \mathtt{ShortCar}) \quad (6)$$

which describes the set of trains composed of at least one closed short car. It provides a correct concept definition for ET w.r.t. the given examples, *i.e.*, the following concept equivalence axiom

$$\mathtt{ET} \equiv\ \exists\, \mathtt{hasCar}.(\mathtt{ClosedCar} \sqcap \mathtt{ShortCar}) \quad (7)$$

is a solution for the CI-CSP problem in hand.

3.2 Metaquerying in Knowledge Graph Mining

In [21] it has been observed that an interesting alternative to language bias (*i.e.*, the declarative bias used in, *e.g.*, [30] to learn rules of a predefined form) is the use of a meta-querying language that could take advantage of some useful meta-information about the data to be analyzed, for instance, the schema of the KG when available. In [17,18] I have proposed a new approach to KG Mining which adapts the notion of metaquery introduced by [29] for DM in relational databases to the novel context of KGs. In particular, a metaquery for KG Mining is a second-order DL conjunctive query under the Henkin semantics. However, the resulting metaquery language can be implemented with standard technologies of the Web of Data such as SPARQL.[4]

Example 4. An example of a metaquery in this context is the following

$$MQ_1 :\ mq(Q, Y, Z) \leftarrow P(X, Y), Q(X, Z) \quad (8)$$

which looks for the properties (Q) holding for the individuals Y. Note that P, Q are higher-order variables whereas X, Y, Z are first-order variables.

[4] https://www.w3.org/TR/sparql11-overview/

Metaqueries can be extended into implications, called *metaquery extensions*, of the form

$$MQ_1 \rightarrow MQ_2 \tag{9}$$

which are actually a compact representation of two metaqueries, MQ_1 and MQ_2, where MQ_2 is longer than - we say *extends* - MQ_1. A shorter notation for (9) is the following which stresses how MQ_2 extends MQ_1

$$MQ_1 \Rightarrow (MQ_2 \setminus MQ_1) \tag{10}$$

The left-hand side and the right-hand side of (10) are called the *body* and the *head* of the metaquery extension, respectively. Note that in the case of query extensions, the head does not correspond to the conclusion (as with clauses). Following the standard terminology, one should rather bear in mind the unshortened notation, and call MQ_2 the *conclusion* of the metaquery extension. Metaquery extensions serve as a template for rules we are interested in when applying rule mining algorithms to a given KG.

Example 5. Let us consider the following metaquery

$$MQ_2: \; mq(Q, Y, Z) \leftarrow P(X, Y), Q(X, Z), Q(Y, Z) \tag{11}$$

which looks for the properties (Q) holding for the individuals Y and shared with the individuals X to which Y is related by some P. From (8) and (11) we can build a metaquery extension as shown below

$$P(X, Y), Q(X, Z) \Rightarrow Q(Y, Z) \tag{12}$$

with reference to the KG depicted in Fig. 2, (1) is an instantiation of (12) obtained by substituting the variables P and Q with the role names $isMarriedTo$ and $livesIn$, respectively.

4 Final remarks

The work summarized in this paper pursues an interesting direction of research at the intersection of ML/DM and KR. For this research I have taken inspiration from recent results in both areas, notably De Raedt *et al.*'s work on declarative modeling for ML/DM [6], Colucci *et al.*'s work on non-standard reasoning in DLs [4] and Gaggl *et al.*'s proposal of a fixed-domain semantics for DLs [8]. Interestingly, the former two works pursue a unified view on the inferential problems of interest to the respective fields of research. This match of research efforts in the two fields has motivated the work presented in [16] with the aim of bridging the gap between KR and ML/DM in areas such as the maintenance of knowledge bases (or graphs) where the two fields have already produced promising results though mostly independently from each other. New questions and challenges have then been raised by the cross-fertilization of these results. Notably, the choice of a solver is a critical issue, which was more recently addressed in [19,20]. Finally, and from a broader perspective, the work here summarized contributes to the current shift in AI from programming to solving as recently argued by Geffner [11]. However, much work is still to be done.

As for the use of metamodeling in Concept Learning, I plan to implement and test the approach by relying on available tools. Besides empirical evaluation, I intend also to investigate how to express optimality criteria such as the information gain function within the second-order concept expressions. Linking the approach to existing work on ontologies for ML/DM problems is another interesting direction of future research.

As for the use of metaquerying in Knowledge Graph Mining, several aspects of the proposed approach should to be clarified before an implementation. First, I plan to better define the semantics for the proposed metaquery language, also concerning the link with SPARQL. Second, I intend to design algorithms for the instantiation stage and choose the most appropriate evaluation measures for the intended application.

Acknowledgements This work was partially funded by the INdAM - GNCS Project 2019 “Metodi per il trattamento di incertezza ed imprecisione nella rappresentazione e revisione di conoscenza”.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation and Applications (2nd ed.). Cambridge University Press (2007)
2. Badea, L., Nienhuys-Cheng, S.: A refinement operator for description logics. In: Cussens, J., Frisch, A. (eds.) Inductive Logic Programming, Lecture Notes in Artificial Intelligence, vol. 1866, pp. 40–59. Springer-Verlag (2000)
3. Cohen, W.W., Hirsh, H.: Learnability of description logics. In: Haussler, D. (ed.) Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992. ACM (1992)
4. Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F.M., Ragone, A.: A unified framework for non-standard reasoning services in description logics. In: Coelho, H., Studer, R., Wooldridge, M. (eds.) ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings. Frontiers in Artificial Intelligence and Applications, vol. 215, pp. 479–484. IOS Press (2010)
5. De Giacomo, G., Lenzerini, M., Rosati, R.: Higher-order description logics for domain metamodeling. In: Burgard, W., Roth, D. (eds.) Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011 (2011)
6. De Raedt, L.: Declarative modeling for machine learning and data mining. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part I. Lecture Notes in Computer Science, vol. 7523, pp. 2–3. Springer (2012). https://doi.org/10.1007/978-3-642-33460-3_2

7. De Raedt, L.: Languages for learning and mining. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. pp. 4107–4111 (2015), http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9934
8. Gaggl, S.A., Rudolph, S., Schweizer, L.: Fixed-domain reasoning for description logics. In: Kaminka, G.A., Fox, M., Bouquet, P., Hüllermeier, E., Dignum, V., Dignum, F., van Harmelen, F. (eds.) ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016). Frontiers in Artificial Intelligence and Applications, vol. 285, pp. 819–827. IOS Press (2016). https://doi.org/10.3233/978-1-61499-672-9-819
9. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast rule mining in ontological knowledge bases with AMIE+. VLDB Journal **24**(6), 707–730 (2015), https://doi.org/10.1007/s00778-015-0394-1
10. Gebser, M., Leone, N., Maratea, M., Perri, S., Ricca, F., Schaub, T.: Evaluation techniques and systems for answer set programming: a survey. In: Lang, J. (ed.) Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden. pp. 5450–5456. ijcai.org (2018). https://doi.org/10.24963/ijcai.2018/769
11. Geffner, H.: Artificial intelligence: From programs to solvers. AI Communications **27**(1), 45–51 (2014). http://dx.doi.org/10.3233/AIC-130581
12. Guns, T., Nijssen, S., De Raedt, L.: Itemset mining: A constraint programming perspective. Artificial Intelligence **175**(12-13), 1951–1983 (2011)
13. Konev, B., Lutz, C., Ozaki, A., Wolter, F.: Exact learning of lightweight description logic ontologies. J. Mach. Learn. Res. **18**, 201:1–201:63 (2017), http://jmlr.org/papers/v18/16-256.html
14. Lehmann, J., Hitzler, P.: Concept learning in description logics using refinement operators. Machine Learning **78**(1-2), 203–250 (2010)
15. Lenzerini, M., Lepore, L., Poggi, A.: Answering metaqueries over hi (OWL 2 QL) ontologies. In: Kambhampati, S. (ed.) Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016. pp. 1174–1180. IJCAI/AAAI Press (2016), http://www.ijcai.org/Abstract/16/170
16. Lisi, F.A.: A declarative modeling language for concept learning in description logics. In: Riguzzi, F., Zelezny, F. (eds.) Inductive Logic Programming, 22nd International Conference, ILP 2012, Dubrovnik, Croatia, September 17-19, 2012, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7842. Springer Berlin Heidelberg (2013)
17. Lisi, F.A.: Towards a metaquery language for mining the web of data. In: Calì, A., Wood, P.T., Martin, N.J., Poulovassilis, A. (eds.) Data Analytics - 31st British International Conference on Databases, BICOD 2017, London, UK, July 10-12, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10365, pp. 90–93. Springer (2017). http://dx.doi.org/10.1007/978-3-319-60795-5_8
18. Lisi, F.A.: Mining the web of data with metaqueries. In: Riguzzi, F., Bellodi, E., Zese, R. (eds.) Up-and-Coming and Short Papers of the 28th International Conference on Inductive Logic Programming (ILP 2018), Ferrara, Italy, September 2-4, 2018. CEUR Workshop Proceedings, vol. 2206, pp. 92–99. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2206/paper8.pdf
19. Lisi, F.A.: Model with dls + solve with asp! - A case study from concept learning. In: Fiorentini, C., Momigliano, A. (eds.) Proceedings of the 31st Italian Conference on Computational Logic, Milano, Italy, June 20-22, 2016. CEUR Workshop

Proceedings, vol. 1645, pp. 174–189. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1645/paper_12.pdf
20. Lisi, F.A.: A model+solver approach to concept learning. In: Adorni, G., Cagnoni, S., Gori, M., Maratea, M. (eds.) AI*IA 2016: Advances in Artificial Intelligence - XVth International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, November 29 - December 1, 2016, Proceedings. Lecture Notes in Computer Science, vol. 10037, pp. 266–279. Springer (2016), http://dx.doi.org/10.1007/978-3-319-49130-1_20
21. Lisi, F.A., Stepanova, D.: Combining rule learning and nonmonotonic reasoning for link prediction in knowledge graphs. In: Bassiliades, N., Bikakis, A., Costantini, S., Franconi, E., Giurca, A., Kontchakov, R., Patkos, T., Sadri, F., Woensel, W.V. (eds.) Proceedings of the Doctoral Consortium, Challenge, Industry Track, Tutorials and Posters @ RuleML+RR 2017 hosted by International Joint Conference on Rules and Reasoning 2017 (RuleML+RR 2017), London, UK, July 11-15, 2017. CEUR Workshop Proceedings, vol. 1875. CEUR-WS.org (2017), http://ceur-ws.org/Vol-1875/paper20.pdf
22. Michalski, R.: Pattern recognition as a rule-guided inductive inference. IEEE transactions on Pattern Analysis and Machine Intelligence **2**(4), 349–361 (1980)
23. Mitchell, T.M.: Generalization as search. Artificial Intelligence **18**, 203–226 (1982)
24. Motik, B.: On the properties of metamodeling in OWL. Journal of Logic and Computation **17**(4), 617–637 (2007)
25. Muggleton, S.H.: Inductive logic programming. In: Arikawa, S., Goto, S., Ohsuga, S., Yokomori, T. (eds.) Proceedings of the 1st Conference on Algorithmic Learning Theory. Springer/Ohmsma (1990)
26. Muggleton, S.H.: Meta-interpretive learning: Achievements and challenges (invited paper). In: Costantini, S., Franconi, E., Woensel, W.V., Kontchakov, R., Sadri, F., Roman, D. (eds.) Rules and Reasoning - International Joint Conference, RuleML+RR 2017, London, UK, July 12-15, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10364, pp. 1–6. Springer (2017). https://doi.org/10.1007/978-3-319-61252-2_1
27. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. Proceedings of the IEEE **104**(1), 11–33 (2016). http://dx.doi.org/10.1109/JPROC.2015.2483592
28. Pan, J.Z., Horrocks, I.: OWL FA: a metamodeling extension of OWL DL. In: Carr, L., De Roure, D., Iyengar, A., Goble, C.A., Dahlin, M. (eds.) Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006. pp. 1065–1066. ACM (2006)
29. Shen, W., Ong, K., Mitbander, B.G., Zaniolo, C.: Metaqueries for data mining. In: Advances in Knowledge Discovery and Data Mining, pp. 375–398. AAAI/MIT Press (1996)
30. Tran, H.D., Stepanova, D., Gad-Elrab, M.H., Lisi, F.A., Weikum, G.: Towards nonmonotonic relational learning from knowledge graphs. In: Cussens, J., Russo, A. (eds.) Inductive Logic Programming - 26th International Conference, ILP 2016, London, UK, September 4-6, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10326, pp. 94–107. Springer (2017). http://dx.doi.org/10.1007/978-3-319-63342-8_8