KI - Künstliche Intelligenz
https://doi.org/10.1007/s13218-020-00650-1

SYSTEMS DESCRIPTION

NoHR: An Overview

Reasoning with Ontologies and Nonmonotonic Rules

Vedran Kasalica[1] · Matthias Knorr[2] · João Leite[2] · Carlos Lopes[3]

Received: 3 January 2020 / Accepted: 26 February 2020

Abstract
Description logic ontologies, such as ontologies written in OWL, and non-monotonic rules, as known in Logic Programming, are two major approaches in Knowledge Representation and Reasoning. Even though their integration is challenging due to their inherent differences, the need to combine their distinctive features stems from real world applications. In this paper, we give an overview of NoHR, a reasoner designed to answer queries over theories composed of an OWL ontology in a Description logic and a set of non-monotonic rules. NoHR has been developed as a plug-in for the widely used ontology editor Protégé, building on a combination of reasoners dedicated to OWL and rules, but it is also available as a library, allowing for its integration within other environments and applications. It comes with support for all polynomial OWL profiles and the integration of their constructors as well as for standard built-in Prolog predicates, and allows the direct consultation of databases during query evaluation and the usage of sophisticated mechanisms, such as tabling already computed results, all of which enhances the applicability and the efficiency of query answering.

Keywords Query answering · Description logic ontologies · Rule-based languages

1 Introduction

Ontology languages founded on Description Logics (DLs) [1] and non-monotonic rule languages as established in Logic Programming (LP) [2] are both major formalisms in Knowledge Representation and Reasoning (KRR), each with its own distinct benefits and features. As a matter of fact, the standardization of the Semantic Web driven by the W3C[1] incorporates today the Web Ontology Language (OWL) [3] and the Rule Interchange Format (RIF) [4].

Ontology languages are widely used to represent and reason over taxonomic knowledge. As decidable fragments of first-order logic, they are monotonic by nature, which means that drawn conclusions persist when acquiring new information. Also, they allow reasoning on abstract information, such as relations between classes of objects, even without knowing any concrete instances. The balance between expressiveness and complexity of reasoning with ontology languages is crucial for DLs, which is why the very expressive general language OWL 2, with its high worst-case complexity, includes three tractable (polynomial) profiles [5] each with a different application purpose in mind.

Non-monotonic rules focus on reasoning over instances, explicitly representing inference from premises to conclusions. Commonly, the Closed World Assumption (CWA) is applied, i.e., the absence of a piece of information suffices to derive it being false, until new information to the

Partially supported by FCT projects RIVER (PTDC/CCI-COM/30952/2017) and NOVA LINCS (UIDB/04516/2020).

✉ Matthias Knorr
mkn@fct.unl.pt

Vedran Kasalica
v.kasalica@uu.nl

João Leite
jleite@fct.unl.pt

Carlos Lopes
cml13098@campus.fct.unl.pt

[1] Department of Information and Computing Sciences, Utrecht University, 3584 CC Utrecht, The Netherlands

[2] Departamento de Informática, NOVA LINCS, FCT-NOVA Lisboa, 2829-516 Caparica, Portugal

[3] Centre of Technology and Systems, UNINOVA, FCT-NOVA Lisboa, 2829-516 Caparica, Portugal

[1] http://www.w3.org.

Published online: 04 March 2020

contrary is provided, hence being non-monotonic. Thus, they are well-suited to declaratively model defaults and exceptions, in the sense that the absence of an exceptional feature can be used to derive that the (more) common case applies, and also integrity constraints, which can be used to ensure that the data under consideration conforms to the desired specifications.

Though the combination of both formalisms is non-trivial due to the mismatch between the semantic assumptions of the two formalisms and the considerable differences as to how decidability is ensured in each of them (see, e.g., [6–8]), it has frequently been requested by applications [9–13]. For example, in clinical health care, large ontologies such as SNOMED CT,[2] that are captured by the OWL 2 profile OWL 2 EL and its underlying description logic (DL) $\mathcal{EL}^{++}$ [14], are used for electronic health record systems, clinical decision support systems, or remote intensive care monitoring, to name only a few. Yet, expressing conditions such as dextrocardia, i.e., that the heart is exceptionally on the right side of the body, requires non-monotonic rules. Another example can be found in [9], where modeling pharmacy data of patients with the closed-world assumption would have been preferred in the study to match patient records with clinical trials criteria, because usually it can be assumed that a patient is not under a specific medication unless explicitly known. Also, in [10] it is shown that in Legal Reasoning, besides the well-known need for default reasoning afforded by non-monotonic rules, it is also necessary to reason in the absence of concrete known individuals (instances), hence requiring features found in ontology languages such as DLs. Moreover, in the risk assessment of cargo shipments, where ontologies are used to model, e.g., the international taxonomy of goods, and rules are required to model, among others, the inspection policies of shipments, another application scenario can be found [11, 12]. In addition, maintaining a telecommunications inventory is another use case [13], where an ontology is used to represent the hierarchy of existing equipment (for different countries), and rules to model and identify different failures to be detected.

In these cases, we often have to deal with large knowledge bases and/or large quantities of data as well as ontologies that are covered by the constructors the polynomial OWL 2 profiles provide, but do not necessarily fit one profile, and query answering needs to be highly efficient.

In this paper, we describe the latest version of NoHR[3] (Nova Hybrid Reasoner) that allows the user to query combinations of ontologies and non-monotonic rules in a top-down manner, which means that only the part of the ontology and rules that is relevant for the query is actually evaluated. It comes as a plug-in for the ontology editor Protégé 5.X,[4] – in fact, the first hybrid reasoner of its kind for Protégé – but it is also available as a library, allowing for its integration within other environments and applications.

NoHR is theoretically founded on the formalism of Hybrid MKNF under the well-founded semantics [15] which comes with two main arguments in its favor (cf. the related work in [6, 7] on combining DLs with non-monotonic rules). First, the overall approach, introduced in [7] and based on the logic of minimal knowledge and negation as failure (MKNF) [16], provides a very general and flexible framework for combining DL ontologies and non-monotonic rules (see [7]). Second, [15], which is a variant of [7] based on the well-founded semantics [17] for logic programs, has a lower data complexity than the former – it is polynomial for polynomial DLs – and is amenable for applying top-down query procedures, such as **SLG**($\mathcal{O}$) [18], to answer queries based only on the information relevant for the query, and without computing the entire model – no doubt a crucial feature when dealing with large ontologies and huge amounts of data.

NoHR is implemented in a way that combines the capabilities of the DL reasoners ELK [19], HermiT [20], and Konclude [21] with the rule engine XSB Prolog,[5] and provides native support for Databases. The latter is encoded through the concept of mappings between predicates (both in the rules and the ontology) and SQL query results from the corresponding database systems, which is realized using ODBC drivers, thus allowing the integration of NoHR with all major database management systems. In addition, NoHR exhibits the following features:[6]

- Support for ontologies written in any of the three tractable OWL 2 Profiles, and for those combining the constructors permitted in these profiles;
- Support for a vast number of standard built-in Prolog predicates;
- Mapping editor and rule editor within Protégé accompanied with a rule parser;
- Possibility to define predicates with arbitrary arity in Protégé;
- Guaranteed termination of query answering;
- Robustness w.r.t. inconsistencies between the ontology and the rules (see, e.g., [22]);
- Scalable fast interactive response times.

[2] http://www.ihtsdo.org/snomed-ct/.

[3] http://nohr.di.fct.unl.pt.

[4] http://protege.stanford.edu.

[5] http://xsb.sourceforge.net.

[6] The source code can be obtained at https://github.com/NoHRReasoner/NoHR.

2 Hybrid Knowledge Bases

We first present an overview on the kind of hybrid knowledge bases considered for NoHR, starting with some information on the ontologies we consider.

Description logics (DLs)[7] are commonly decidable fragments of first-order logic, defined over disjoint countably infinite sets of *concept names* $\mathsf{N_C}$, *role names* $\mathsf{N_R}$, and *individual names* $\mathsf{N_I}$, matching unary and binary predicates, and constants, respectively. *Complex concepts* (and *complex roles*) can be defined based on these sets and the logical constructors a concrete DL admits to be used. An *ontology* $\mathcal{O}$ is a finite set of *inclusion axioms* of the form $C \sqsubseteq D$ where C and D are both (complex) concepts (or roles) and *assertions* of the form $C(a)$ or $R(a, b)$ for concepts C, roles R, and individuals a, b. The semantics of such ontologies is defined in a standard way for first-order logic.

The DL $\mathcal{SROIQ}$ [23] underlying the W3C standard OWL 2 is very general and highly expressive, but reasoning with it is highly complex, which is why the profiles OWL 2 EL, OWL 2 QL and OWL 2 RL have been defined [5], for which reasoning is tractable. NoHR supports all three profiles, in fact, even a combination of the constructors provided by them [12]. Here, we give a brief overview on important features as supported in NoHR.

First, $\mathcal{EL}^+_\bot$, a large fragment of $\mathcal{EL}^{++}$ [14], the DL underlying the tractable profile OWL 2 EL [5], only allows conjunction of concepts, existential restriction of concepts ($\exists R.C$ for roles R and concepts C basically corresponding to $\forall x \exists y R(x, y) \wedge C(y)$), hierarchies of roles, and disjoint concepts. $\mathcal{EL}^+_\bot$ is tailored towards reasoning with large conceptual models, i.e., large TBoxes.

DL-Lite$_R$, one language of the *DL-Lite* family [24] and underlying OWL 2 QL, admits in addition role inverses and disjoint roles, but in exchange only simple role hierarchies, no conjunction, and limitations on the use of existential restrictions in particular on the left hand side of inclusion axioms. This profile focuses on answering queries over huge amounts of data, and is amenable to the usage of relational database technology.

Finally, Description Logic Programs [25], which underly OWL 2 RL, have been introduced with the aim to find a fragment that can be directly translated into rules (and vice-versa), and therefore can also be implemented using a rule reasoner. Due to that, this profile allows more constructors than the other two profiles, but usually restricts their usage to one side of the inclusion axioms to ensure that the translation to rules is possible.

Many ontologies matching one of these profiles exist and are used in applications, but there are also those that do use features from more than one of those, such as the bio-ontology Galen[8] or the famous benchmark ontology LUBM.[9] For these, reasoning is no longer polynomial, but highly efficient general purpose OWL reasoners nevertheless make their usage feasible in practice.

The hybrid knowledge bases we consider here are based on MKNF knowledge bases (KBs), which build on the logic of minimal knowledge and negation as failure (MKNF) [16]. Among the two different semantics defined for these [7, 15], we focus on the well-founded one [15], due to its lower computational complexity and amenability to top-down querying without computing the entire model. We only point out important notions, and refer to [15, 18] for the details.

A *rule* r is of the form $H \leftarrow A_1, \ldots, A_n, \mathbf{not}\ B_1, \ldots, \mathbf{not}\ B_m$ where the *head* of r, H, and all A_i with $1 \leq i \leq n$ and B_j with $1 \leq j \leq m$ in the *body* of r are atoms, possibly built from the unary and binary predicates occurring in the ontology.[10] A *program* $\mathcal{P}$ is a finite set of rules, $\mathcal{O}$ is an ontology, and an *MKNF knowledge base* $\mathcal{K}$ is a pair $(\mathcal{O}, \mathcal{P})$. A rule r is *safe* if all its variables occur in at least one A_i with $1 \leq i \leq n$, and $\mathcal{K}$ is *safe* if all its rules are safe.[11]

The semantics of MKNF knowledge bases $\mathcal{K}$ is given by a translation π into an MKNF formula $\pi(\mathcal{K})$, i.e., a formula over first-order logic extended with two modal operators **K** and **not**. The well-founded MKNF model can be computed efficiently [15] in a bottom-up fashion, and queried based on **SLG**($\mathcal{O}$), as defined in [18]. This procedure extends SLG resolution with tabling [26] with an *oracle* to $\mathcal{O}$ that handles ground queries to the DL-part of $\mathcal{K}$ by returning (possibly empty) sets of atoms that, together with $\mathcal{O}$ and information already proven true, allows us to derive the queried atom. We refer to [18] for the full account of **SLG**($\mathcal{O}$).

To tackle the integration of several databases within an MKNF KB, we apply mappings.[12] Essentially, mappings are used to create predicates that are populated with the result set obtained from queries to external databases, which also allows us to consult tables from different databases. A

[7] We refer to [1] for a more general and thorough introduction to DLs.

[8] https://bioportal.bioontology.org/ontologies/GALEN.

[9] http://swat.cse.lehigh.edu/projects/lubm/.

[10] Conceptually, this allows to simultaneously view certain predicates under the closed world semantics in rules and under the open world semantics in the ontology, and admits the bidirectional flow of information between both the rules and the ontology.

[11] In general, the notion of DL-safety is used in this context which requires that these variables occur in atoms that do themselves not occur in the ontology, but due to the reasoning method employed in NoHR, we can relax that restriction.

[12] Similar concepts have been used before for adding database support to rule systems, such as *DLV*^{DB} [27], and in ontology based data access, such as in ontop [28].

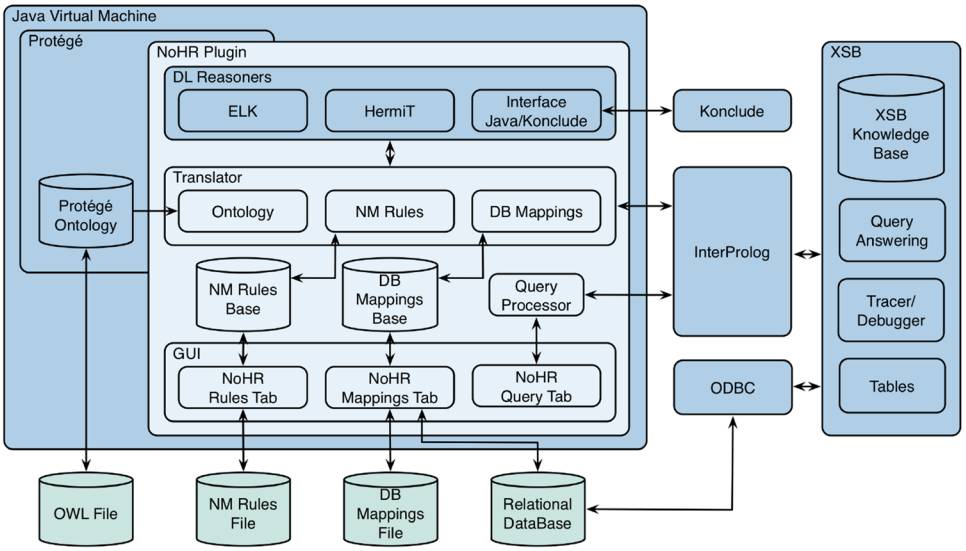

Fig. 1 System architecture of NoHR v4.0 with native database support

mapping for predicate p is a triple $\langle p, db, q\rangle$ where the result set from db for the query q (defined over db) is mapped to the predicate p. Based on this, the hybrid knowledge bases we consider are triples $\mathcal{K} = (\mathcal{O}, \mathcal{P}, \mathcal{M})$, where $\mathcal{O}$ is an ontology, $\mathcal{P}$ is a finite set of rules, and $\mathcal{M}$ is a set of mappings. In terms of semantics, the integration can be achieved by extending the translation function to mappings and the mapping knowledge base by turning, for each triple $\langle p, db, q\rangle$, all tuples for p into rule facts.

3 Architecture

In this section, we describe the architecture of NoHR v4.0, and discuss several features of its implementation.

NoHR is available as a plugin for Protégé[13], a well-known and widely used ontology editor, and we describe the system architecture of this plugin NoHR v4.0 as shown in Fig. 1.

The input for the plugin consists of an OWL file, a rule file and a mappings file. All three components can be edited in Protégé, using the built-in interface for the ontology and the custom "NoHR Rules" and "NoHR Mappings" tabs, provided by the plugin, for the rule and mapping components. The former (as well as the query panel) comes with a dedicated parser to support the creation of correctly formed rules and queries. The latter allows the creation of mappings based on the user's specification of what columns from which tables of which database should be combined, where the underlying SQL queries are dynamically generated, based on the structure of the schema, which allows the automatic application of several optimizations to the generated queries. Alternatively, arbitrary SQL queries can be written to take advantage of the capabilities of the specific DBMS at hand for the sake of, e.g., benefiting from using advanced joins and the associated performance gains when querying.

After the inputs (which can be empty) and the first query are provided, the ontology is translated into a set of rules, using one of the provided reasoners, ELK [19], HermiT [20] or Konclude [21], depending on the DL in which the ontology is written. This resulting set of rules is not equivalent to the ontology in general, but it yields exactly the same answers for ground queries (for more details cf. [12, 29, 30]). The resulting set is then combined with the rules and mappings provided by the input. This joined result serves as input to XSB Prolog via InterProlog,[14] which is an open-source Java front-end, allowing the communication between Java and a Prolog engine, and the query is sent via the same interface to XSB to be executed. During the execution, mappings are providing facts from the external databases as they are requested in the reasoning process. This procedure is supported by the installed ODBC connections and handled within XSB, thus providing full control over the database access during querying and taking advantage of the built-in optimization to access only the relevant part of the database. Answers are returned to the query processor, which displays them to the user in a table (in the Query Tab). The user may pose further queries, and the system will simply send them directly to XSB, without any repeated preprocessing. If the knowledge base is edited, the system recompiles only the component that was changed.

[13] https://protege.stanford.edu/.

[14] http://interprolog.com/java-bridge/.

4 Evaluation

The different versions of NoHR have been evaluated with a focus on different aspects [12, 13, 29, 30], and we summarize the main observations here.

First, different ontologies can be preprocessed for querying in a short amount of time (around one minute for SNOMED CT with over 300,000 concepts), and increasing the number of rules only raises the time for translation linearly. This is suitable, as the preprocessing commonly only needs to be done once before querying. Notably, we compared how the preprocessing period is affected by the usage of the different reasoners using standard ontologies of different expressiveness. As shown in [12], ELK is indeed always fastest, which confirms that by default, whenever the ontology fits $\mathcal{EL}^+_\bot$, the dedicated translation module should be used. Additionally, we observe that HermiT is faster than Konclude in all instances where it does not time out, which justifies using HermiT whenever it is capable of classifying the given ontology. Still, Konclude turns out to be useful, in the cases where HermiT fails to classify an ontology that does not fit the OWL 2 EL profile.

It was also observed that querying time is in general neglectable, in comparison to preprocessing. In particular, we have shown that (i) NoHR scales reasonably well for OWL QL query answering without non-monotonic rules (only slowing down for memory-intensive cases), (ii) preprocessing is even faster for OWL QL and RL when compared to OWL EL (or their generalization), because, no prior classification is necessary, (iii) querying scales well, even for over a million facts/assertions in the ABox, despite being slightly slower on average for OWL QL in comparison to to the other cases, as part of the OWL inferences is encoded in the rule translations directly, and (iv) adding rules scales linearly for pre-processing and querying, even for an ontology with many negative inclusions (for *DL-Lite*$_R$).

In addition, with respect to the database component, we have shown that, if the data is stored in a database and accessed directly during querying instead of being loaded into memory in the form of facts or ontology assertions, preprocessing time and memory consumption substantially reduces, in particular for tuples of higher arity. In terms of querying, on average querying becomes slightly slower, as the connection via ODBC adds an overhead to the query process. However, if we use advanced mappings, which allow us to outsource certain joins over data from XSB to the DBMS, then improvements of considerable margin can be achieved, in particular when advanced database joins reduce the amount of data that needs to be sent to XSB for reasoning.

5 Conclusions

We have provided an overview on NoHR, a reasoner which allows us to query knowledge bases composed of both an ontology in one of the OWL 2 profiles and even a union of their language features and a set of non-monotonic rules. It is available as a plugin for Protégé as well as an API and uses a top-down reasoning approach, which means that only the part of the ontology and rules that is relevant for the query is actually evaluated. It comes with support for databases, robustness to inconsistencies, and fast interactive response times (after a brief preprocessing period), even for larger knowledge bases.

In terms of future work, adapting ideas on semi-automatic mapping creation [27] is promising to further improve usability. Also, a comparison with the HEX formalism [31, 32], based on dl-programs [6], is of interest (even if arguably less general than hybrid MKNF [7]), in particular w.r.t. the integration of databases with ontologies and rules as well as novel evaluation techniques [33]. Likewise, the work on Clopen Knowledge Bases deserves further inspection [34] in that regard. Establishing closer relations to a generalization in nonmonotonic description logics [35] are also worth considering.

A more ambitious objective is the integration of data on the Semantic Web [36], in particular in Linked Open Data, an initiative which has gained considerable interest and has already been applied in diverse areas (see, e.g., [37, 38]). While, conceptually, the idea corresponds to database integration, the technical solution will certainly differ, due to different standards and formats employed. In addition, turning the focus to dynamic hybrid KB's [11, 39–42] in the presence of streams [43, 44] to a setting with heterogeneous knowledge bases [45–47] (building on established connections between these and hybrid knowledge bases [48]) is a particularly interesting road for future research. Given the wide-spread availability of Linked Open Data sets nowadays and the vast amounts of data in many applications in the form of streams, such additions would provide a valuable extension to NoHR for knowledge integration.

Acknowledgements We would like to thank the anonymous reviewers for the helpful comments and thank Nuno Costa and Vadim Ivanov for their contributions to the development of NoHR.

References

1. Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF (eds) (2010) The description logic handbook: theory, implementation, and applications, 3rd edn. Cambridge University Press, Cambridge

2. Baral C, Gelfond M (1994) Logic programming and knowledge representation. J Log Program 19(20):73–148
3. Hitzler P, Krötzsch M, Parsia B, Patel-Schneider PF, Rudolph S (2012) OWL 2 web ontology language: primer. W3C, Cambridge
4. Kifer M, Boley H (eds.): RIF overview (Second Edition). W3C Working Group Note (2013) Available at http://www.w3.org/TR/rif-overview/. Accessed 5 Feb 2013
5. Motik B, Cuenca Grau B, Horrocks I, Wu Z, Fokoue A, Lutz C (eds) (2012) OWL 2 web ontology language: profiles, 2nd edn. W3C, Cambridge
6. Eiter T, Ianni G, Lukasiewicz T, Schindlauer R, Tompits H (2008) Combining answer set programming with description logics for the semantic web. Artif Intell 172(12–13):1495–1539
7. Motik B, Rosati R (2010) Reconciling description logics and rules. J ACM 57(5):93–154
8. Eiter T, Simkus M (2015) Linking open-world knowledge bases using nonmonotonic rules. In: Ianni G, Truszczynski M, Calimeri F (eds) Procs. of LPNMR, LNCS, vol 9345. Springer, Berlin, pp 294–308
9. Patel C et al (2007) Matching patient records to clinical trials using ontologies. In: Aberer K et al (eds) Semantic web. ISWC 2007, ASWC 2007. Lecture notes in computer science, vol 4825. Springer, Berlin, pp 816–829
10. Alberti M, Knorr M, Gomes AS, Leite J, Gonçalves R, Slota M (2012) Normative systems require hybrid knowledge bases. In: van der Hoek W, Padgham L, Conitzer V, Winikoff M et al (eds) Procs. of AAMAS. IFAAMAS, Montreal pp, pp 1425–1426
11. Slota M, Leite J, Swift T (2015) On updates of hybrid knowledge bases composed of ontologies and rules. Artif Intell 229:33–104
12. Lopes C, Knorr M, Leite J (2017) Nohr: Integrating XSB prolog with the OWL 2 profiles and beyond. Procs of LPNMR, LNCS, vol 10377. Springer, Berlin, pp 236–249
13. Kasalica V, Gerochristos I, Alferes JJ, Gomes AS, Knorr M, Leite J (2019) Telco network inventory validation with nohr. In: Lierler Y, Woltran S, Balduccini M (eds) Procs. of LPNMR, LNCS, vol 11481. Springer, Berlin, pp 18–31
14. Baader F, Brandt S, Lutz C (2005) Pushing the *EL* envelope. In: Kaelbling LP, Saffiotti A (eds) Procs. of IJCAI. Professional Book Center, pp. 364–369
15. Knorr M, Alferes JJ, Hitzler P (2011) Local closed world reasoning with description logics under the well-founded semantics. Artif Intell 175(9–10):1528–1554
16. Lifschitz V (1991) Nonmonotonic databases and epistemic queries. In: Mylopoulos J, Reiter R (eds) Procs. of IJCAI. Morgan Kaufmann, Burlington
17. Gelder AV, Ross KA, Schlipf JS (1991) The well-founded semantics for general logic programs. J ACM 38(3):620–650
18. Alferes JJ, Knorr M, Swift T (2013) Query-driven procedures for hybrid MKNF knowledge bases. ACM Trans Comput Log 14(2):1–43
19. Kazakov Y, Krötzsch M, Simančík F (2013) The incredible ELK: from polynomial procedures to efficient reasoning with *EL* ontologies. J Autom Reason 53:1–61
20. Glimm B, Horrocks I, Motik B, Stoilos G, Wang Z (2014) Hermit: an OWL 2 reasoner. J Autom Reason 53(3):245–269
21. Steigmiller A, Liebig T, Glimm B (2014) Konclude: system description. J Web Sem 27:78–85
22. Kaminski T, Knorr M, Leite J (2015) Efficient paraconsistent reasoning with ontologies and rules. In: Yang Q, Wooldridge MJ (eds) Procs. of IJCAI. AAAI Press, Louisiana, pp 3098–3105
23. Horrocks I, Kutz O, Sattler U (2006) The even more irresistible *SROIQ*. In: Doherty P, Mylopoulos J, Welty CA (eds) Procs. of KR. AAAI Press, Louisiana, pp 57–67
24. Artale A, Calvanese D, Kontchakov R, Zakharyaschev M (2009) The *DL-Lite* family and relations. J Artif Intell Res (JAIR) 36:1–69
25. Grosof BN, Horrocks I, Volz R, Decker S (2003) Description logic programs: combining logic programs with description logic. In: Hencsey G, White B, Chen YR, Kovács L, Lawrence S (eds) Procs. of WWW. ACM, New York, pp 48–57
26. Chen W, Warren DS (1996) Tabled evaluation with delaying for general logic programs. J ACM 43(1):20–74
27. Terracina G, Leone N, Lio V, Panetta C (2008) Experimenting with recursive queries in database and logic programming systems. TPLP 8(2):129–165
28. Calvanese D, Cogrel B, Komla-Ebri S, Kontchakov R, Lanti D, Rezk M, Rodriguez-Muro M, Xiao G (2017) Ontop: answering SPARQL queries over relational databases. Semantic Web 8(3):471–487
29. Ivanov V, Knorr M, Leite J (2013) A query tool for *EL* with nonmonotonic rules. In: Alani H et al (eds) The semantic web—ISWC 2013. ISWC 2013. Lecture notes in computer science, vol 8218. Springer, Berlin, pp 216–231
30. Costa N, Knorr M, Leite J (2015) Next step for NoHR: OWL 2 QL. In: Arenas M, Corcho Ó, Simperl E, Strohmaier M, d'Aquin M, Srinivas K, Groth PT, Dumontier M, Heflin J, Thirunarayan K, Staab S (eds.) Procs. of ISWC, *LNCS*, vol. 9366, pp. 569–586
31. Eiter T, Fink M, Ianni G, Krennwallner T, Redl C, Schüller P (2016) A model building framework for answer set programming with external computations. TPLP 16(4):418–464
32. Eiter T, Germano S, Ianni G, Kaminski T, Redl C, Schüller P, Weinzierl A (2018) The DLVHEX system. KI 32(2–3):187–189
33. Redl C (2019) Inlining external sources in answer set programs. TPLP 19(3):360–411
34. Bajraktari L, Ortiz M, Simkus M (2018) Combining rules and ontologies into clopen knowledge bases. In: McIlraith SA, Weinberger KQ (eds) Procs. of AAAI. AAAI Press, Louisiana, pp 1728–1735
35. Knorr M, Hitzler P, Maier F (2012) Reconciling OWL and nonmonotonic rules for the semantic web. In: Raedt LD, Bessiere C, Dubois D, Doherty P, Frasconi P, Heintz F, Lucas PJF (eds) Procs. of ECAI, frontiers in artificial intelligence and applications, vol 242. IOS Press, Amsterdam, pp 474–479
36. Glimm B, Stuckenschmidt H (2016) 15 years of semantic web: an incomplete survey. KI 30(2):117–130
37. Latif A, Scherp A, Tochtermann K (2016) LOD for library science: benefits of applying linked open data in the digital library setting - retrospects and research topics. KI 30(2):149–157
38. Zapilko B, Schaible J, Wandhöfer T, Mutschke P (2016) Applying linked data technologies in the social sciences. KI 30(2):159–162
39. Slota M, Leite J (2010) Towards closed world reasoning in dynamic open worlds. TPLP 10(4–6):547–563
40. Slota M, Leite J, Swift T (2011) Splitting and updating hybrid knowledge bases. TPLP 11(4–5):801–819
41. Slota M, Leite J (2012) A unifying perspective on knowledge updates. In: del Cerro LF, Herzig A, Mengin J (eds) Procs of JELIA, LNCS, vol 7519. Springer, Berlin, pp 372–384
42. Slota M, Leite J (2014) The rise and fall of semantic rule updates based on se-models. TPLP 14(6):869–907
43. Beck H, Dao-Tran M, Eiter T (2018) LARS: a logic-based framework for analytic reasoning over streams. Artif Intell 261:16–70
44. Beck H, Dao-Tran M, Eiter T, Folie C (2018) Stream reasoning with LARS. KI 32(2–3):193–195
45. Brewka G, Ellmauthaler S, Gonçalves R, Knorr M, Leite J, Pührer J (2018) Reactive multi-context systems: heterogeneous reasoning in dynamic environments. Artif Intell 256:68–104

46. Brewka G, Ellmauthaler S, Pührer J (2014) Multi-context systems for reactive reasoning in dynamic environments. In: Schaub T, Friedrich G, O'Sullivan B (eds) Procs of ECAI, frontiers in artificial intelligence and applications, vol 263. IOS Press, Amsterdam, pp 159–164
47. Gonçalves R, Knorr M, Leite J (2014) Evolving multi-context systems. In: Friedrich G, O'Sullivan B (eds) Procs. of ECAI, frontiers in artificial intelligence and applications, vol 263. IOS Press, Amsterdam, pp 375–380
48. Knorr M, Slota M, Leite J, Homola M (2014) What if no hybrid reasoner is available? hybrid MKNF in multi-context systems. J Log Comput 24(6):1279–1311