Description Logic reasoning for Semantic web ontologies

– Extended abstract

Anni-Yasmin Turhan
Institute for Theoretical Computer Science
Technische Universität Dresden
Dresden, Germany
turhan@tcs.inf.tu-dresden.de

1. INTRODUCTION

The ontology language for the semantic web OWL provides means to describe entities of an application domain in an ontology in a well-structured way. The underlying formalism for OWL are Description Logics (DLs) [6], which are a family of knowledge representation formalisms that have formal semantics. This family of logics is tailored towards representing terminological knowledge of an application domain in a structured and formally well-understood way.

Description logics allow users to define important notions, such as classes or relations of their application domain in terms of concepts and roles. These concepts (unary predicates) and roles (binary predicates) then restrict the way these classes and relations are interpreted. Based on these definitions, implicitly captured knowledge can be inferred from the given descriptions of concepts and roles. These inferences are defined based on the formal semantics of DLs. A great range of these inferences has been defined and investigated for a variety of DLs with differing expressivity, where the expressivity of a DL is determined by the means it allows to describe concepts and roles. The inferences devised for DLs are, for instance, sub-class or instance relationships. The investigation of algorithms for reasoning services and their complexity is the main focus of the DL research community. Mainly during the nineties reasoning procedures for more and more expressive DLs have been investigated. Naturally, the gain in expressiveness came at the cost of higher complexity for the reasoning procedures—reasoning for the DLs investigated is PSpace-complete or even ExpTime-complete [33, 28, 49] (for an overview see [9, 16]).

Despite the high complexity, highly optimized DL reasoning systems were implemented based on the tableau method—most prominently the FaCT system [24] and Racer [21]. These systems employed optimization methods developed for DL reasoning based on tableaux [7, 23, 32, 22] and demonstrated that the high worst case complexities would hardly be encountered in practice [24, 32, 20, 25] In fact, it turned out that these highly optimized implementations of the reasoning methods do perform surprisingly well on DL knowledge bases from practical applications. Encouraged by these findings and driven by application needs researchers investigated tableau algorithms for even more expressive DLs [29, 30, 26, 31] in the last decade, which were implemented in tableau-based reasoners for very expressive DL such as Pellet [45], FaCT++ [50, 51] and RacerPro [44] More recently, new reasoning methods for expressive DLs were investigated and implemented such as resolution [37, 39] in KAON2 and hyper-tableau [40, 41, 19] in HermiT.

Another line of research is dedicated to the design of so-called *light-weight DLs*, which are DLs with relatively limited expressivity, but good computational properties for specific reasoning tasks [8]. Reasoning even for very large ontologies written in these DLs can be done efficiently, since the respective reasoning methods are tractable. There are two "families" of lightweight DLs: the $\mathcal{EL}$ family [12, 4, 5], for which the subsumption and the instance problem are polynomial, and the DL Lite family [13, 15], for which the instance problem and query answering are polynomial. A member of each of these families is the DL corresponding to one of the profiles of the OWL 2 standard.

DLs have been employed in various domains, such as databases, biomedical or context-aware applications [3, 48]. Their most notable success so far is probably the adoption of the DL-based language OWL as standard ontology language for the Semantic Web [27].

In the following we give pointers to the relevant literature for the topics that that were covered in the tutorial.

2. DL PRELIMINARIES

Historically, DLs stem from knowledge representation systems such as *semantic networks* [43, 46] or *frame systems* [36]. These early knowledge representation systems were motivated by linguistic applications and offer methods to compute inheritance relations between the specified notions from the domain of discourse. These early knowledge representation systems have operational semantics, i.e., the semantics of reasoning is given by its implementation. As a consequence, the result of the reasoning process depends on the implementation of the reasoner [47]. To remedy this, DLs and their reasoning services are based on formal semantics. The information about the application domain is represented in a declarative and unambiguous way. More importantly, the formal semantics of the reasoning services ensure behavior of the DL reasoning systems independent of their implementation. For comprehensive introductions on DLs we refer the reader to [6, 2, 52]

The central notion for DLs are *concept descriptions*, which can be built from concept names and so-called *concept constructors*. For instance, one can describe a tutorial as an event given by a lecturer in the following way by a concept description:

$$\mathsf{Event} \sqcap \exists\, \mathsf{given\text{-}by}.\mathsf{Lecturer} \sqcap \exists\, \mathsf{has\text{-}topic}.\top$$

WIMS'11 May 25-27, 2011 Sogndal, Norway

Table 1: OWL syntax, DL syntax and semantics of concept descriptions.

constructor name	OWL syntax	DL syntax	semantics
conjunction	intersectionOf	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	someValuesFrom	$\exists r.C$	$\{x \in \Delta \mid \exists y : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
value restriction	allValuesFrom	$\forall r.C$	$\{x \in \Delta \mid \forall y : (x,y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
negation	complementOf	$\neg C$	$\Delta \setminus C^{\mathcal{I}}$
disjunction	unionOf	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
qualified number restrictions	qualified cardinality restrictions	$(\leq\ n\ r\ C)$ $(\geq\ n\ r\ C)$	$\{x \in \Delta_{\mathcal{I}} \mid \#\{y : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \leq n\}$ $\{x \in \Delta_{\mathcal{I}} \mid \#\{y : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \geq n\}$

This concept description is a conjunction ($\sqcap$) of the concept Event, the existential restriction $\exists$ given-by.Lecturer and the existential restriction $\exists$ has-topic.$\top$. The first existential restriction consists of the role name given-by and concept Lecturer, which relates the Lecturer to the tutorial. The latter existential restriction states that there is a topic (which is not specified).

In general, concept descriptions are built from the set of concept names N_C and the set of role names N_R using concept constructors. Every DL offers a different set of concept constructors. The DL $\mathcal{EL}$ allows only for the concept constructors that were used in the example concept description above. We call concept descriptions of the form $\exists r.C$ existential restrictions and concept descriptions of the form $C \sqcap D$ conjunctions. The semantics of DL concept descriptions is given by means of interpretations.

Let C and D be $\mathcal{EL}$-concept descriptions and r a role name. An interpretation is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where the *domain* $\Delta^{\mathcal{I}}$ is a non-empty set and $\cdot^{\mathcal{I}}$ is a function that assigns to every concept name A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every role name r a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. This function is extended to complex $\mathcal{EL}$-concept descriptions as follows:

- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$;
- $(\exists r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid$ there is a $y \in \Delta^{\mathcal{I}}$ with $(x,y) \in r^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\}$; and

Most DLs also offer the *top-concept* $\top$, which is always interpreted as the whole domain $\Delta^{\mathcal{I}}$. Now, with the $\mathcal{EL}$-concept constructors at hand, one can, for instance, characterize a graduate CS student by the following concept description:

$\exists$ studies. CS $\sqcap$ (Master-Student $\sqcap$ has-degree. Bachelor)

Concept description like these are the main building blocks to model terminological knowledge. The DL defined above is called $\mathcal{EL}$ and is the core DL of the $\mathcal{EL}$-family.

Depending on the set of concept constructors on can define more expressive DLs. The concept constructors given in Table 1 form the DL $\mathcal{ALCQ}$.

For expressive DLs, roles can have declared properties or can be related to other roles. A role r can be declared to be a *transitive role* in the TBox. The semantics is straight-forward. An interpretation $\mathcal{I}$ satisfies a transitive role declaration transitive(r) if $\{(a,b),(b,c)\} \subseteq r^{\mathcal{I}}$ implies $(a,c) \in r^{\mathcal{I}}$. The declaration of an *inverse role* applies to a role name r and yields its inverse r^{-1}, where the semantics is the obvious one, i.e., $(r^{-1})^{\mathcal{I}} := \{(e,d) \mid (d,e) \in r^{\mathcal{I}}\}$. Furthermore, it can be specified that a role is a super-role of another role by a *role inclusion axiom*. The set of all role inclusions form the *role hierarchy*. An interpretation $\mathcal{I}$ satisfies a role inclusion axiom $r \sqsubseteq s$ if $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$. The DL that offers all the concept constructors and role declarations introduced is the DL $\mathcal{SHIQ}$. This DL is the logic underlying OWL DL.

DL knowledge bases are typically divided two parts. The

Terminological part , which captures the characterization of the basic categories from an application, and the

Assertional part , which captures the facts from the application.

We introduce these components of DL knowledge bases now.

2.1 Terminological Knowledge

A name can be assigned to a concept description by a *concept definition*. For instance, we can write Tutorial $\equiv$ Event $\sqcap$ $\exists$ given-by.Lecturer $\sqcap$ $\exists$ has-topic.$\top$ to supply a concept definition for the concept Tutorial. Let A be a concept name and C, D be (possibly) complex concept description.

- A *concept definition* is a statement of the form $A \equiv C$.
- A *general concept inclusion* (GCI for short) is a statement of the form $C \sqsubseteq D$.

It is easy to see that every concept definition $A \equiv C$ can be expressed by two GCIs: $A \sqsubseteq C$ and $C \sqsubseteq A$. The terminological information expressed by GCIs is collected in the so-called *TBox*, which is simply a finite set of GCIs. An interpretation is a model of a TBox $\mathcal{T}$, if it satisfies all GCIs, i.e., if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all $C \sqsubseteq D$ in $\mathcal{T}$.

If all concept descriptions in a TBox $\mathcal{T}$ are from a description logic $\mathcal{L}$, then we call $\mathcal{T}$ a $\mathcal{L}$-TBox.

If a concept definition $A \equiv C$ in a TBox uses a concept name B directly, i.e., B appears in C, or if B is used indirectly by the definitions of the names appearing in C, we say that the TBox is *cyclic*. Otherwise a TBox is *acyclic*.

2.2 Assertional Knowledge

Individual facts from the application domain can be stated by *assertions*. There are two basic kinds of assertions for DL systems—one expresses that an individual belongs to a concept and the other one specifies that two individuals are related via a role. The set N_I is the set of all individual names. Let C be a (possibly complex) concept description, $r \in NR$ a role name and i, j ($\{i,j\} \subseteq N_I$) be two individual names, then

- $C(i)$ is called a *concept assertion* and
- $r(i,j)$ is called a *role assertion*.

An ABox $\mathcal{A}$ is a finite set of concept assertions and role assertions.

For instance, we can express that Dresden is a city located at the river Elbe by the following ABox:

{ City(Dresden), River(Elbe), located-at(Dresden, Elbe) }

If all concept descriptions in an ABox $\mathcal{A}$ are from a Description Logic $\mathcal{L}$, then we call $\mathcal{A}$ a $\mathcal{L}$-ABox.

In order to capture ABoxes, the interpretation function is now extended to individual names. Each individual name is mapped by the interpretation function to an element of the domain $\Delta^{\mathcal{I}}$.

Let C be a concept description, r a role name and i, j two individual names, then an interpretation $\mathcal{I}$ satisfies

- the concept assertion $C(i)$ if $i^{\mathcal{I}} \in C^{\mathcal{I}}$ and
- the role assertion $r(i, j)$ if $(i^{\mathcal{I}}, j^{\mathcal{I}}) \in r^{\mathcal{I}}$.

An interpretation $\mathcal{I}$ is a model of an ABox $\mathcal{A}$, if $\mathcal{I}$ satisfies every assertion in $\mathcal{A}$. A DL *knowledge base* $\mathcal{K}$ consists of an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$. We write $\mathcal{K}$ = ($\mathcal{T}$, $\mathcal{A}$). We now discuss standard reasoning services for DLs, which are implemented in most DL systems.

2.3 Reasoning services

One of the basic reasoning services in DL systems is to test for the *satisfiability* of a concept or a TBox, i.e., to test whether the information specified in it contains logical contradictions or not. In case the TBox contains a contradiction, any consequence can follow logically from the TBox. Moreover, if a TBox is not satisfiable, the specified information can hardly capture the intended meaning from an application domain. To test for satisfiability is often a first step for a user to check whether a TBox models something "meaningful".

Let C be a concept description and $\mathcal{T}$ a TBox. The concept description C is *satisfiable* iff it has a model, i.e., iff there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$. A TBox $\mathcal{T}$ is *satisfiable* iff it has a model, i.e., an interpretation that satisfies all GCIs in $\mathcal{T}$.

If a concept or a TBox is not satisfiable, it is called *unsatisfiable*.

Other typical reasoning services offered in DL systems test for equivalence or inclusion relations between concepts. In the latter case, if one concept of the TBox models a more general category than another one, we say that this concept *subsumes* the other one.

Let C, D be two concept descriptions and $\mathcal{T}$ a (possibly empty) TBox. The concept description C is *subsumed* by the concept description D w.r.t. $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$), iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds in every model $\mathcal{I}$ of $\mathcal{T}$. Two concepts C, D are *equivalent w.r.t.* $\mathcal{T}$ ($C \equiv_{\mathcal{T}} D$), iff $C^{\mathcal{I}} = D^{\mathcal{I}}$ holds for every model $\mathcal{I}$ of $\mathcal{T}$.

A test for the equivalence of concept descriptions can be carried out by two subsumption tests, since the following holds: ($C \equiv_{\mathcal{T}} D$), iff $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$.

The computation of the subsumption relations for *all named concepts* mentioned in the TBox $\mathcal{T}$ is called *classification* of the TBox $\mathcal{T}$ and yields the *concept hierarchy* of the TBox $\mathcal{T}$.

Similarly as for concept descriptions and TBoxes, we can test for the absence of contradictions in ABoxes. An ABox $\mathcal{A}$ is *consistent* w.r.t. a TBox $\mathcal{T}$, iff it has a model that is also a model for $\mathcal{T}$. The individual i is an *instance of* the concept description C w.r.t. an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ (we write $\mathcal{A} \models_{\mathcal{T}} C(i)$), iff $i^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$. The reasoning service *ABox realization* computes for each individual i of an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ the set of all named concepts A appearing in $\mathcal{A}$ and $\mathcal{T}$ that (1) have i as an instance ($\mathcal{A} \models_{\mathcal{T}} A(i)$) and (2) that is least w.r.t. $\sqsubseteq_{\mathcal{T}}$.

With the help of the instance service, one can also compute answers to *instance queries*, i.e., all individuals occurring in the ABox that are instances of the query concept C. In order to state more general search criteria, one can use so-called *conjunctive queries*, i.e., conjunctions of assertions that may also contain variables, of which some can be existentially quantified. For example, the conjunctive query

$$\exists x, y.\mathsf{City}(x) \wedge \mathsf{River}(y) \wedge \ \mathsf{located\text{-}at}(x, y)$$

asks for all Cities that located at a river. With respect to the ABox we have introduced earlier, this conjunctive query has the pair of individuals: Dresden, Elbe as an answer.

2.4 Reasoning Techniques

There are three main reasoning approaches for the DLs that underlie OWL. For the expressive DLs, which offer all Boolean concept constructors, most reasoning services can be reduced to consistence of an ABox w.r.t. a TBox in polynomial time. In presence of full negation we can devise the following *polynomial time* reductions.

- Equivalence can be be reduced to subsumption:
  $C \equiv_{\mathcal{T}} D$ iff $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$.
- Subsumption can be be reduced to (un)satisfiability:
  $C \sqsubseteq_{\mathcal{T}} D$ iff $C \sqcap \neg D$ is unsatisfiable w.r.t. $\mathcal{T}$.
- Satisfiability can be be reduced to consistency:
  C is satisfiable w.r.t. $\mathcal{T}$ iff the ABox $\{C(a)\}$ is consistent w.r.t. $\mathcal{T}$.
- The instance problem can be reduced to (in)consistency:
  $\mathcal{A} \models_{\mathcal{T}} C(a)$ iff $\mathcal{A} \cup \{\neg C(a)\}$ is inconsistent w.r.t. $\mathcal{T}$.

As a consequence, only one reasoning procedure needs to be investigated and implemented in systems in order to use all reasoning services in practice. In fact, all reasoning systems for expressive DLs use this approach and implement the tableaux method for the consistency test. For a more detailed discussion see [2, 52]. Even query answering can be implemented by the tableau method [18, 34, 42, 17]. However, the reduction-based approach requires the presence of negation in the DL.

For light-weight DLs this constructor is not available. Thus different methods for reasoning have been devised. In case of the $\mathcal{EL}$-family, the completion method [12, 4, 5] actually classifies the given TBox $\mathcal{T}$, i.e., it simultaneously computes all subsumption relationships between the concept names occurring in $\mathcal{T}$. This algorithm proceeds in four steps:

1. Normalize the TBox.
2. Translate the normalized TBox into a graph.
3. Complete the graph using completion rules.
4. Read off the subsumption relationships from the normalized graph.

By the completion method a TBox can be classified in polynomial time. This method is implemented in the $\mathcal{EL}$-reasoner jCEL [35].

The other lightweight family of DLs is the DL-Lite family, which is tailored to perform query answering efficiently [14, 15]. Here the approach is to translate the initial query such that it also captures the relevant information from the TBox and then use a relational query engine to answer the translated query. For a comprehensive description of the method see [8]. This method for query answering is implemented in the system QuOnto [1].

3. THE STANDARD ONTOLOGY LANGUAGE OWL

The first version of the OWL standard [27, 10] covered three dialects of increasing expressivity. While the most expressive dialect *OWL full* is beyond the expressivity of DLs and reasoning in it is undecidable, the other two dialects correspond to DLs for which sound and complete reasoning procedures exist. *OWL DL* can express ontologies written in the DL $\mathcal{SHOIQ}$ and the less expressive *OWL lite* can express ontologies written in the DL $\mathcal{SHIN}$.

The close relation of DLs and these two OWL dialects raised the interest of new user groups in DL reasoning and DL systems on the one hand and it helped to develop reasoning tools for broader user communities on the other hand.

OWL DL, the standard ontology language for the Semantic Web, is based on an expressive DL for which reasoning is highly intractable. Its sublanguage OWL Lite was intended to provide a tractable version of OWL, but turned out to be only of a slightly lower worst-case complexity than OWL DL itself. This has led to the development of two new families of light-weight DLs, $\mathcal{EL}$ and *DL-Lite*, which have been proposed as profiles of OWL 2 [53, 38], the new version of the OWL standard. The OWL 2 El profile yields the DL $\mathcal{EL}^{++}$, for which polynomial subsumption algorithms are known [12, 4, 5]. The OWL 2 QL profile corresponds to DL-Lite$_{\mathcal{R}}$, which belongs to the DL-Lite family of DLs. These DLs are tailored towards applications in which huge amounts of data (represented as an ABox) are queried w.r.t. fairly light-weight ontologies. Conjunctive queries can be answered even in LogSpace regarding data complexity [14], i.e. the size of the ABox.

4. REFERENCES

[1] A. Acciarri, D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, M. Palmieri, and R. Rosati. QuOnto: Querying ontologies. In M. M. Veloso and S. Kambhampati, editors, *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI'05)*, pages 1670–1671, 2005. QuOnto download page `http://www.dis.uniroma1.it/~quonto/index.htm`.

[2] F. Baader. Description logics. In *Proceedings of Reasoning Web: Semantic Technologies for Information Systems*, volume 5689 of *Lecture Notes in Computer Science*, pages 1–39, 2009.

[3] F. Baader, A. Bauer, P. Baumgartner, A. Cregan, A. Gabaldon, K. Ji, K. Lee, D. Rajaratnam, and R. Schwitter. A novel architecture for situation awareness systems. In M. Giese and A. Waaler, editors, *Proceedings of the 18th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (Tableaux 2009)*, volume 5607 of *Lecture Notes in Computer Science*, pages 77–92. Springer-Verlag, 2009.

[4] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*, Edinburgh, UK, 2005. Morgan-Kaufmann Publishers.

[5] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope further. In K. Clark and P. F. Patel-Schneider, editors, *In Proc. of the OWLED Workshop*, 2008.

[6] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[7] F. Baader, E. Franconi, B. Hollunder, B. Nebel, and H.-J. Profitlich. An empirical analysis of optimization techniques for terminological representation systems or: Making KRIS get a move on. *Applied Artificial Intelligence. Special Issue on Knowledge Base Management*, 4:109–132, 1994.

[8] F. Baader, C. Lutz, and A.-Y. Turhan. Small is again Beautiful in Description Logics. *KI – Künstliche Intelligenz*, 24(1):25–33, April 2010.

[9] F. Baader and U. Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69:5–40, 2001.

[10] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL web ontology language reference. W3C Recommendation, February 2004. `http://www.w3.org/TR/owl-ref/`.

[11] R. J. Brachman and H. J. Levesque. *Readings in Knowledge Representation*. Morgan Kaufmann, Los Altos, 1985.

[12] S. Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In R. L. de Mantáras and L. Saitta, editors, *Proc. of the 16th European Conf. on Artificial Intelligence (ECAI-04)*, pages 298–302. IOS Press, 2004.

[13] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. DL-Lite: Tractable description logics for ontologies. In M. M. Veloso and S. Kambhampati, editors, *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI'05)*, pages 602–607. AAAI Press/The MIT Press, 2005.

[14] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 260–270, 2006.

[15] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.

[16] D. Calvanese and G. D. Giacomo. Expressive description logics. In *[6]*, pages 178–218. Cambridge University Press, 2003.

[17] T. Eiter, C. Lutz, M. Ortiz, and M. Simkus. Query answering in description logics with transitive roles. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence IJCAI09*. AAAI Press, 2009.

[18] B. Glimm, I. Horrocks, C. Lutz, and U. Sattler. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. In M. M. Veloso, editor, *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI-07)*, pages 399–404, Hyderabad, India, 2007.

[19] B. Glimm, I. Horrocks, and B. Motik. Optimized Description Logic Reasoning via Core Blocking. In J. Giesl and R. Hähnle, editors, *Proc. of the 5th Int. Joint Conf. on Automated Reasoning (IJCAR-10)*, volume 6173 of *Lecture Notes in Computer Science*, pages 457–471, Edinburgh, UK, 2010. Springer.

[20] V. Haarslev and R. Möller. High performance reasoning with very large knowledge bases: A practical case study. In B. Nebel, editor, *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI-01)*, pages 161–166, 2001.

[21] V. Haarslev and R. Möller. RACER system description. In R. Goré, A. Leitsch, and T. Nipkov, editors, *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR-01)*, Lecture Notes in Computer Science. Springer, 2001.

[22] V. Haarslev, R. Möller, and A.-Y. Turhan. Exploiting pseudo models for TBox and ABox reasoning in expressive description logics. In R. Goré, A. Leitsch, and T. Nipkov, editors, *Proceedings of the International Joint Conference on Automated Reasoning IJCAR'01*, LNAI. Springer Verlag, 2001.

[23] I. Horrocks. *Optimising Tableaux Decision Procedures for Description Logics*. PhD thesis, University of Manchester, 1997.

[24] I. Horrocks. Using an expressive description logic: FaCT or fiction? In A. Cohn, L. Schubert, and S. Shapiro, editors, *Proc. of the 6th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-98)*, pages 636–647,

1998.

[25] I. Horrocks. Reasoning with expressive description logics: Theory and practice. In A. Voronkov, editor, *Proc. of the 19th Conf. on Automated Deduction (CADE-19)*, number 2392 in Lecture Notes In Artificial Intelligence, pages 1–15. Springer, 2002.

[26] I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible $\mathcal{SROIQ}$. In P. Doherty, J. Mylopoulos, and C. Welty, editors, *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-06)*, pages 57–67. AAAI Press, 2006.

[27] I. Horrocks, P. 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.

[28] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *Journal of Logic and Computation*, 9(3):385–410, 1999.

[29] I. Horrocks and U. Sattler. Optimised reasoning for $\mathcal{SHIQ}$. In *Proc. of the 15th European Conference on Artificial Intelligence*, 2002.

[30] I. Horrocks and U. Sattler. A tableaux decision procedure for $\mathcal{SHOIQ}$. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*. Morgan Kaufmann, Jan. 2005.

[31] I. Horrocks and U. Sattler. A tableau decision procedure for $\mathcal{SHOIQ}$. *J. of Automated Reasoning*, 39(3):249–276, 2007.

[32] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *J. of the Interest Group in Pure and Applied Logic*, 8(3):239–264, 2000.

[33] C. Lutz. Complexity of terminological reasoning revisited. In *Proc. of the 6th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR'99)*, Lecture Notes in Computer Science, pages 181–200. Springer, 1999.

[34] C. Lutz. The complexity of conjunctive query answering in expressive description logics. In A. Armando, P. Baumgartner, and G. Dowek, editors, *Proceedings of the 4th International Joint Conference on Automated Reasoning (IJCAR2008)*, number 5195 in LNAI, pages 179–193. Springer, 2008.

[35] J. Mendez and B. Suntisrivaraporn. Reintroducing CEL as an OWL 2 EL reasoner. In B. Cuenca Grau, I. Horrocks, B. Motik, and U. Sattler, editors, *Proc. of the 2008 Description Logic Workshop (DL 2009)*, volume 477 of *CEUR-WS*, 2009.

[36] M. Minsky. A framework for representing knowledge. Technical report, MIT-AI Laboratory, Cambridge, MA, USA, 1974.

[37] B. Motik. *Reasoning in Description Logics using Resolution and Deductive Databases*. PhD thesis, Universität Karlsruhe, 2006.

[38] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 web ontology language profiles. W3C Recommendation, 27 October 2009. `http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/`.

[39] B. Motik and U. Sattler. A Comparison of Techniques for Querying Large Description Logic ABoxes. In M. Hermann and A. Voronkov, editors, *Proc. of the 13th Int. Conf. on Logic for Programming Artificial Intelligence and Reasoning (LPAR'06)*, LNCS, Phnom Penh, Cambodia, November 13–17 2006. Springer. KAON2 download page: `http://kaon2.semanticweb.org/`.

[40] B. Motik, R. Shearer, and I. Horrocks. A hypertableau calculus for $\mathcal{SHIQ}$. In D. Calvanese, E. Franconi, V. Haarslev, D. Lembo, B. Motik, S. Tessaris, and A.-Y. Turhan, editors, *Proc. of the 2007 Description Logic Workshop (DL 2007)*, June 2007.

[41] B. Motik, R. Shearer, and I. Horrocks. Optimized Reasoning in Description Logics using Hypertableaux. In F. Pfennig, editor, *Proc. of the 23th Conf. on Automated Deduction (CADE-23)*, LNAI, pages 67–83, Bremen, Germany, July 17–20 2007. Springer.

[42] M. Ortiz, D. Calvanese, and T. Eiter. Data complexity of query answering in expressive description logics via tableaux. *Journal of Automated Reasoning*, 41(1):61–98, 2008.

[43] M. R. Quillian. Word concepts: A theory and simulation of some basic capabilities. *Behavioral Science*, 12:410–430, 1967. Republished in [11].

[44] Racer Systems GmbH & Co. KG. *RacerPro Reference Manual Version 1.9*, Dec. 2005. Available from: `http://www.racer-systems.com/products/racerpro/reference-manual-1-9.pdf`.

[45] E. Sirin and B. Parsia. Pellet system description. In B. Parsia, U. Sattler, and D. Toman, editors, *Description Logics*, volume 189 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006.

[46] J. F. Sowa, editor. *Principles of Semantic Networks*. Morgan Kaufmann, Los Altos, 1991.

[47] J. F. Sowa. *Encyclopedia of Artificial Intelligence*, chapter Semantic Networks. John Wiley & Sons, New York, 1992.

[48] T. Springer and A.-Y. Turhan. Employing description logics in ambient intelligence for modeling and reasoning about complex situations. *Journal of Ambient Intelligence and Smart Environments*, 1(3):235–259, 2009.

[49] S. Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *Journal of Artificial Intelligence Research*, 12:199–217, May 2000.

[50] D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of the 3rd Int. Joint Conf. on Automated Reasoning (IJCAR-06)*, 2006. FaCT++ download page: `http://owl.man.ac.uk/factplusplus/`.

[51] D. Tsarkov, I. Horrocks, and P. F. Patel-Schneider. Optimising terminological reasoning for expressive description logics. *Journal of Automated Reasoning*, 2007.

[52] A.-Y. Turhan. Reasoning and explanation in $\mathcal{EL}$ and in expressive description logics. In U. Aßmann, A. Bartho, and C. Wende, editors, *Reasoning Web*, number 6325 in LNCS, pages 1–27. Springer, 2010.

[53] W3C OWL Working Group. OWL 2 web ontology language document overview. W3C Recommendation, 27th October 2009. `http://www.w3.org/TR/2009/REC-owl2-overview-20091027/`.