

Title: Description Logics
Authors: Adila Krisnadhi, Pascal Hitzler
Affil./Addr.: Wright State University, Kno.e.sis Center
377 Joshi Research Center, 3640 Colonel Glenn Highway, Dayton
OH 45435, USA
Phone: (937)775-5217
E-mail: adila@knoesis.org, pascal@knoesis.org

Description Logics

Glossary

KR: knowledge representation.

DLs: Description Logics; a family of logic-based KR languages for representing knowledge through assertions about concepts, individuals and relationships among them.

(Logic-based) Semantics: a way to interpret any statement in a language; logic-based semantics interprets such a statement using operations in mathematical logic.

Interpretation: a mathematical structure realizing the semantics of a language, typically consisting of an underlying set (domain of interest) and a mapping from the statements in the language to the set or mathematical operations on it.

Model: an interpretation that interprets logical statements in non-contradictory way.

Individual: an element of the domain of interest. Individual names are atomic statements in a DL corresponding to such elements.

Concept: a statement in a DL corresponding to sets of individuals.

Role: a statement in a DL corresponding to a binary relation between individuals.

Axiom: a statement in a DL that asserts certain constraints that have to be satisfied by some concepts, roles and individuals.

TBox: a set of axioms constraining concepts.

ABox: a set of axioms constraining particular individuals.

RBox: a set of axioms constraining roles.

KB: knowledge base; a set of axioms of any kind.

Reasoning: a process in which implicit knowledge/facts are inferred from explicit knowledge given through a set of axioms.

Open-world Assumption: a meta-level semantical assumption in which a statement is considered false only when the KB forces it so. This means a lack of knowledge does not imply falsity.

Definition

Description logics (DLs) [Baader et al 2007, Krötzsch et al 2012] is a family of knowledge representation (KR) languages which represent knowledge in a domain of interest using formal, logic-based semantics through knowledge bases (KBs) containing general assertions describing relevant concepts – hence, the term *description* — and specific assertions about individuals and relationships among them. DLs owe their origin to semantic networks and frame systems. Reasoning that enables implicit knowledge to be inferred from KBs is an indispensable part of DLs and is typically decidable (as opposed to first-order logic which is undecidable). DLs are also prominent as the underlying formalism for the Web Ontology Language (OWL) [Hitzler et al 2009]. Further discussions on reasoning algorithms and OWL, however, are covered under separate titles.

DL Syntax and Semantics

Knowledge is represented through DL KBs which consist of axioms composed from expressions, called *concepts*, *roles*, and the always-atomic *individual names*. Non-atomic concepts and roles are constructed from the atomic ones using various constructors admitted by particular DL in consideration. The semantics is realized via *interpretations*, each is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$, called the *domain*, is a non-empty (possibly infinite) set of individuals, and $\cdot^{\mathcal{I}}$, called the *interpretation function*, maps each atomic concept A to a set of individuals $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each atomic role R to a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and each individual name a to an individual $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The semantics of non-atomic concepts and roles are then obtained by extending the mapping $\cdot^{\mathcal{I}}$ depending on which constructor is used to built them. Table 1 lists the syntax and semantics of some prominent DL constructors.

Table 1. Common DL concept and role constructors where C, D are (possibly non-atomic) concepts, R, S are (possibly non-atomic) roles, a is an individual name, n is a nonnegative integer, and for a set M , $|M|$ is the cardinality of M .

Name	Syntax	Semantics based on an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$
top and concept	\top, \perp	$\Delta^{\mathcal{I}}, \emptyset$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
concept intersection & union	$C \sqcap D, C \sqcup D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}, C^{\mathcal{I}} \cup D^{\mathcal{I}}$
concept complement	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
value/universal restriction	$\forall R.C$	$\{x \mid \forall y: \langle x, y \rangle \in R^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}\}$
existential restriction	$\exists R.C$	$\{x \mid \exists y: \langle x, y \rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$
number (at-least) restriction	$\geq nR.C$	$\{x \mid \{y \mid \langle x, y \rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geq n\}$
number (at-most) restriction	$\leq nR.C$	$\{x \mid \{y \mid \langle x, y \rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leq n\}$
universal role	U	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
inverse role	R^{-}	$\{\langle x, y \rangle \mid \langle y, x \rangle \in R^{\mathcal{I}}\}$
role intersection	$R \sqcap S$	$R^{\mathcal{I}} \cap S^{\mathcal{I}}$

A DL KB consists of a set of axioms that can be categorized as *TBox*, *ABox* and *RBox* axioms. TBox and RBox axioms describe general knowledge about concepts and roles, respectively. On the other hand, ABox axioms describe specific knowledge in the form of membership of an individual in a concept and relationships between individuals through a role. Semantics of axioms are provided as criteria for which they are *satisfied* by an interpretation \mathcal{I} as given in Table 2.

Table 2. Axioms in DLs. Note: A is always a concept name, C, D are concepts, $R_{(i)}, R, S$ are roles.

Name	Syntax	Satisfaction Criteria
concept definition (TBox)	$A \equiv C$	$A^{\mathcal{I}} = C^{\mathcal{I}}$
concept inclusion (TBox)	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
concept assertion (ABox)	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion (ABox)	$R(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
negative role assertion	$\neg R(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \notin R^{\mathcal{I}}$
role equivalence (RBox)	$R \equiv S$	$R^{\mathcal{I}} = S^{\mathcal{I}}$
role hierarchy (RBox)	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
general role inclusion (RBox)	$R_1 \circ \dots \circ R_k \sqsubseteq S$	$R_1^{\mathcal{I}} \circ \dots \circ R_k^{\mathcal{I}} \subseteq S^{\mathcal{I}}$ where 'o' is binary composition of relation
role functionality, transitivity, symmetry, asymmetry, reflexivity, irreflexivity (RBox)	$\text{Fun}(R), \text{Tra}(R), \text{Sym}(R), \text{Asy}(R), \text{Ref}(R), \text{Irr}(R)$	$R^{\mathcal{I}}$ is functional, transitive, symmetric, asymmetric, reflexive, irreflexive

The following simple examples provide better intuition on how DLs are used to represent knowledge. The axiom $\text{Man} \equiv \text{Human} \sqcap \text{Male}$ asserts that a man is precisely a male human. The axiom $\text{Father} \equiv \text{Man} \sqcap \exists \text{hasChild.Human}$ asserts that a father is precisely a man who has a human child. Meanwhile, $\text{Woman} \sqsubseteq \text{Human} \sqcap \neg \text{Man}$ asserts that a woman is a human that is not a man, and $\text{Mother} \equiv \text{Woman} \sqcap \exists \text{hasChild.Human}$ states that a mother is a woman who has a human child. To say that having parent is an inverse relationship of having a child, one can state $\text{hasParent} \sqsubseteq$

$hasChild^-$. Thus, one can also define a child as a human who has either a father or a mother using axiom $Child \sqsubseteq Human \sqcap \exists hasParent.(Father \sqcup Mother)$. Furthermore, the axioms $Son \equiv Child \sqcap Male$ and $Daughter \equiv Child \sqcap \neg Son$ define the concepts son and daughter. If one asserts that a father-with-many-sons must have at least three sons, then this axiom can be used: $FatherWithManySons \sqsubseteq Father \sqcap \geq 3 hasChild.Son$. Meanwhile, the axiom $Father \sqcap \forall hasChild.\neg Son \sqsubseteq FatherWithoutSons$ asserts that a father who has no son is a father-without-sons.

To assert specific statements for particular individuals, one can use ABox assertions. For instance, $FatherWithoutSons(bill)$ asserts that Bill is a father-without-sons. To say that Bill has a child, called Chelsea, one can assert $hasChild(bill, chelsea)$.

Reasoning Tasks for DLs

Based on the semantics of axioms, there are a few basic notions which form the core of reasoning in DLs: satisfiability/consistency checking, subsumption checking and instance checking. First, any set of axioms (including TBoxes, ABoxes, RBoxes, and KBs in general) is *satisfiable* or *consistent* if it has a *model*. A model of a set of axioms is an interpretation \mathcal{I} that *satisfies* all of its axioms. Satisfaction criteria for each type of axiom can be found in Table 2. *KB satisfiability/consistency problem* is thus a reasoning problem of deciding whether a given KB is consistent. One may also be interested in *concept satisfiability* which is deciding whether for a given concept C and a KB, there is a model \mathcal{I} of the KB such that $C^{\mathcal{I}} \neq \emptyset$.

Subsumption checking is a problem of deciding whether a concept C is subsumed by a concept D w.r.t. a KB. This holds when $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model \mathcal{I} of the KB. Note that the subsumption relationship includes not only the ones explicitly stated in the KB (through concept inclusions), but also the ones that can be *inferred* from it. Computing subsumption among all concept names occurring in a KB is called a *classification* which allows one to construct the so-called “is-a” hierarchy if a concept name A is subsumed by a concept name B , then A is below B in the hierarchy (i.e., every (individual in) A is a (individual in) B). The “is-a” hierarchy has also been a key feature in semantic networks and frames.

Instance checking is the problem of deciding whether an individual (name) a belongs to a concept C w.r.t. a KB. This holds when $a^{\mathcal{I}} \in C^{\mathcal{I}}$ in every model \mathcal{I} of the KB. This problem places greater emphasis to ABox axioms due to the involvement of explicit individuals. Instance checking can be generalized to *conjunctive query entailment*: given a KB and a set of expressions $\{C_1(x_1), \dots, C_m(x_m), R_1(y_1, z_1), \dots, R_n(y_n, z_n)\}$ where C_1, \dots, C_m are concepts, R_1, \dots, R_n are roles, and $x_1, \dots, x_m, y_1, \dots, y_n, z_1, \dots, z_n$ are (not necessarily distinct) variables, find a substitution of those variables with individual names occurring in the KB such that the resulting ABox axioms are satisfied by all model of the KB. This problem is closely related to conjunctive query answering which is important considering many real-life situation in which a huge amount of data (which can be seen as an ABox) is augmented with schematic knowledge (in the form of TBox or RBox). Note also the resemblance with notions of query answering from the study of databases, although unlike modeling in databases which are characterized with closed-world assumption and finiteness of the domain, DLs are distinguished with open-world assumption and possible non-finiteness of the domain.

Some Prominent DLs

- \mathcal{ALC} [Baader and Nutt 2007]: simplest Boolean-closed DL that admits top and bottom concept; concept intersection, union, and complement; value and existential restrictions; TBox axioms; and ABox axioms. Reasoning is EXPTIME-complete.
- \mathcal{FL}_0 : simple DL that admits top concept, concept intersection, value restriction, TBox axioms and ABox axioms. It is notable since reasoning is EXPTIME-complete, but polynomial if done with an empty KB [Donini 2007, Baader et al 2005].
- \mathcal{EL} : simple DL that allows top concept, concept intersection, existential restriction, TBox axioms and ABox axioms. It is notable since reasoning is polynomial. In fact, \mathcal{SROEL} (also known as \mathcal{EL}^{++} [Baader et al 2005]), obtained by adding bottom concept, role hierarchy (thus role equivalence), and general role inclusion to \mathcal{EL} is still polynomial. \mathcal{EL}^{++} is adopted for the OWL

2 EL profile of OWL 2 DL standard [Motik et al 2009]. Its sublanguages also found extensive use in biomedical applications.

- *SHIF*: an expressive DL obtained from *ACC* by adding role transitivity, role hierarchy, inverse roles, functionality (concepts of the form $\leq 1R$ and axioms of the form $\text{Fun}(R)$), role symmetry. This DL underlies the OWL 1 Lite standard and has an EXPTIME-complete reasoning [Horrocks and Patel-Schneider 2004, Hayes et al 2004].
- *SHOIN*: very expressive DL, obtained from *SHIF* by adding unqualified number restrictions and nominals. It underlies the OWL 1 DL standard and has an NEXPTIME-complete reasoning [Horrocks and Patel-Schneider 2004, Hayes et al 2004].
- *SROIQ*: very expressive DL, obtained from *SHOIN* by adding general role inclusion, qualified number restriction, role asymmetry, role reflexivity and role irreflexivity. It underlies OWL 2 DL and has an N2EXPTIME-complete reasoning [Horrocks et al 2006, Kazakov 2008, W3C OWL Working Group 2009].

Relationships with Other Formalisms

As a logic-based formalism, DLs are related to many other formalisms [Sattler et al 2007]. Many of these correspondences were in fact exploited to derive complexity results and reasoning algorithms.

Notably, most DLs are a decidable fragment of first-order predicate logic (FOL) with equality. In fact, many DLs are expressible in either \mathcal{L}^k (FOL over unary and binary predicates with at most k variables) or \mathcal{C}^k (like \mathcal{L}^k , but allows counting quantifiers). In the translation, concepts are translated into FOL formulas with one free variable in which concept names correspond to unary predicate, role names to binary predicate and individual names to constants. This relationship also extends to the rule (i.e., Horn) fragment of FOL. Development of Description Logic Programs [Grosz et al 2003] which is roughly an intersection between DLs and binary Datalog rules has resulted in the OWL 2 RL profile of OWL 2 DL. Many other formalisms have also been proposed to realize integration between rule languages and DLs.

DLs are also strongly related to modal logics [Baader and Lutz 2007]. For example, *ACC* can be seen as a notational variant of multi-modal logic \mathbf{K}_m in which concept names correspond to propositional letters, while value and existential restrictions correspond to the modal operators \Box and \Diamond . Other close relationships with modal logic families have also been noted, e.g., with propositional dynamic logics, hybrid logics and guarded fragments.

Relationships with object-oriented and database modeling languages have also been observed. For example, Entity Relationship (ER) models can be translated into DL KBs which enables formally checking for inconsistency. Parts of Unified Modeling Language (UML) specification can also be translated into DL KBs. On the other hand, the need for more expressive query languages over relational database has led to the development of DL-Lite, a family of very simple DLs which can be used to perform very efficient queries. Due to this reason, DL-Lite [Calvanese et al 2007] has been adopted to underlie the OWL 2 QL profile of OWL 2 DL standard.

Cross-references

OWL, Reasoning

Acknowledgements

- Both authors acknowledge the support of the National Science Foundation (NSF) project “TRON – Tractable Reasoning with Ontologies” under the award 1017225 III: Small.
- The first author acknowledges the support of Fulbright Indonesia Presidential Scholarship PhD Grant. 2010–2013

References

- Baader F, Lutz C (2007) Description logic. In: Blackburn P, van Benthem J, Wolter F (eds) Handbook of Modal Logic, Studies in Logic and Practical Reasoning, vol 3, Elsevier, chap 13, pp 757–819
- Baader F, Nutt W (2007) Basic Description Logics, chap 2. In: Baader et al [2007]
- Baader F, Brandt S, Lutz C (2005) Pushing the EL envelope. In: the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05), Morgan-Kaufmann Publishers, Edinburgh, UK
- Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF (eds) (2007) The Description Logic Handbook: Theory, Implementation, and Applications, 2nd edn. Cambridge University Press
- Calvanese D, de Giacomo G, Lembo D, Lenzerini M, Rosati R (2007) Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J of Automated Reasoning* 39(3):385–429
- Donini FM (2007) Complexity of Reasoning, chap 3. In: Baader et al [2007]
- Groszof BN, Horrocks I, Volz R, Decker S (2003) Description logic programs: Combining logic programs with description logic. In: the 12th Int. World Wide Web Conference (WWW 2003), pp 48–57
- Hayes P, Patel-Schneider PF, Horrocks I (2004) OWL web ontology language semantics and abstract syntax. W3C recommendation, W3C, <http://www.w3.org/TR/owl-semantics/>. Accessed on Oct 18, 2012.
- Hitzler P, Krötzsch M, Rudolph S (2009) Foundations of Semantic Web Technologies. Chapman & Hall/CRC
- Horrocks I, Patel-Schneider PF (2004) Reducing OWL entailment to description logic satisfiability. *J of Web Semantics* 1(4):345–357
- Horrocks I, Kutz O, Sattler U (2006) The even more irresistible SROIQ. In: the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006), AAAI Press, pp 57–67
- Kazakov Y (2008) RIQ and SROIQ are harder than SHOIQ. In: the 11th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2008), AAAI Press, pp 274–284
- Krötzsch M, Simančík F, Horrocks I (2012) A description logic primer. CoRR abs/1201.4089
- Motik B, Fokoue A, Horrocks I, Wu Z, Lutz C, Grau BC (2009) OWL 2 web ontology language profiles. W3C recommendation, W3C, <http://www.w3.org/TR/owl2-profiles/>. Accessed on Oct 18, 2012.
- Sattler U, Calvanese D, Molitor R (2007) Relationships with other Formalisms, chap 4. In: Baader et al [2007]
- W3C OWL Working Group (2009) OWL 2 web ontology language document overview. W3C recommendation, W3C, <http://www.w3.org/TR/owl2-overview/>. Accessed on Oct 18, 2012.

Recommended Reading

- [Baader et al 2007] is the standard text for DLs; covers almost all major results in DLs, written in semi-textbook style; requires some basics in mathematical logic.
- [Krötzsch et al 2012] is a text intended as a very first reading on DLs without requiring formal logic background.
- [Hitzler et al 2009] is an introductory level textbook in semantic web technologies which also covers significant amount of DLs material, especially in the context of their application in the Semantic Web.