# On the correspondence between description logics and logics of programs (position paper)

**Giuseppe De Giacomo and Maurizio Lenzerini**
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italia
e-mail: [degiacom,lenzerini]@assi.dis.uniroma1.it

## 1 Introduction

In the last decade, many efforts have been devoted to an analysis of the epistemological adequacy, and the computational effectiveness of Description Logics (DLs). In particular, starting with [4], the research on the computational complexity of the reasoning tasks associated with DLs has shown that in order to ensure decidability and/or efficiency of reasoning in all cases, one must renounce to some of the expressive power [17; 19; 20; 12; 13; 11]. These results have led to a debate on the trade-off between expressive power of representation formalisms and worst-case efficiency of the associated reasoning tasks. This issue has been one of the main themes in the area of DLs, and has led to at least four different approaches to the design of knowledge representation systems.

- In the first approach, the main goal of a DL is to offer powerful mechanisms for structuring knowledge, as well as sound and complete reasoning methods (not necessarily realized by means of terminating procedures), while little attention has to be paid to the (worst-case) computational complexity of the reasoning procedures. Systems like OMEGA [1], LOOM [18] and KL-ONE [6], can be considered as following this approach.
- The second approach advocates a careful design of the DLs so as to offer as much expressive power as possible while retaining the possibility of sound, complete, and efficient (often polynomial in the worst case) inference procedures. Much of the research on CLASSIC [5] follows this approach.
- The third approach, similarly to the first one, advocates very expressive languages, but, in order to achieve efficiency, accepts incomplete reasoning procedures. There is no general consensus on what kind of incompleteness is acceptable. Perhaps, the most interesting attempts are those resorting to a non-standard semantics for characterizing the form of incompleteness [22; 3; 14].
- Finally, the fourth approach is based on what we can call "the expressiveness and decidability thesis", and aims at defining DLs that are both very expressive and decidable, i.e. designed in such a way that sound, complete, and terminating procedures exist for the associated reasoning tasks. Great attention is given in this approach to the complexity analysis for the various sublogics, so as to devise suitable optimization techniques and to single out tractable subcases. This approach is the one followed in the design of KRIS [2].

This position paper presents an ongoing research project that adheres to the fourth approach, and aims at both identifying the most expressive DLs with decidable associated decision problems, and characterizing the computational complexity of the corresponding reasoning problems.

## 2 The expressiveness and decidability thesis

In order to clearly characterize the expressiveness and decidability thesis, let us point out that by "very expressive DL" we mean the following:

1. The logic offers powerful mechanisms to describe/render classes.
   - It includes concept constructs for boolean connectives $C \sqcap D$, $C \sqcup D$, $\neg C$, and existential and universal quantifications $\forall R.C$, $\exists R.C$.
   - It may include role constructs for inverse role $R^-$, chaining of roles $R \circ Q$, union of roles $R \sqcup Q$, and the identity role projected on a concept $id(C)$. It may also include *functional restrictions*[1] on atomic roles $(\leq 1\ P)$ and possibly on their inverse $(\leq 1 P^-)$, and (qualified) *number restrictions*[2] again on atomic roles $(\leq\ n\ P)$, $(\geq\ n\ P)$ $((\leq\ n\ P.C), (\geq\ n\ P.C))$ and possibly on their inverse $(\leq\ n\ P^-)$, $(\geq\ n\ P^-)$ $((\leq\ n\ P^-.C), (\geq\ n\ P^-.C))$.
   - It possibly includes suitable mechanisms to describe concepts which are not first-order definable. The most common example of such

[1] Functional restrictions impose the functionality of a role in the context of a concept.

[2] (Qualified) number restrictions state the minimum and/or the maximum number of instances of a role (restricted by means of concept) in the context of a concept.

mechanisms is a construct for the transitive closure of roles $R^*$. More sophisticated ones are those to capture *inductively* and *co-inductively* defined classes (i.e. classes defined as the smallest class such that ..., or the biggest class such that ...).

2. The logic provides suitable means for imposing mutual dependencies among concepts (TBox). The basic mechanisms for supporting this feature are *inclusion assertions* of the form $C \sqsubseteq D$ where $C, D$ can be any concepts, stating that $C$ is to be interpreted as a subset of $D$. Observe that the assumption of acyclicity of TBoxes is not enforced. Indeed, with this assumption, the power of inclusion assertions vanishes.

3. The logic allows one to assert properties of individuals (ABox) in term of *membership assertions*. These can be of two forms: $a : C$, stating that an individual $a$ is an instance of the concept $C$, and $a\ R\ b$, stating that the individual $a$ is related via the role $R$ to the individual $b$.

Note that, the presence of inverse of roles allows the logic to subsume most of the frame-based representation systems, semantic data models and object-oriented database models proposed in the literature. The constructs for functional restrictions on both atomic roles and their inverse greatly enhance the power of the logic, e.g. they allow the logic to correctly represent n-ary relations among classes. Note, also, that the ability to describe non-first-order definable classes is often needed, for example to model the most common data structures used in computer science, such as LISTs and TREEs.[3]

## 3 The correspondence between DLs and logics of programs

Two main approaches have been developed following the "expressiveness and decidability thesis". The first approach relies on the tableaux-based technique proposed in [25; 12], and led to the identification of a decision procedure for a logic which fully covers points (2) and (3) above, but only partially point (1) in that it does not include the construct for inverse roles [7], and has no mechanism to describe concepts that are not first-order definable.

The second approach is based on the work by Schild [23], which singled out a correspondence between some DLs of the kind described above and a certain class of logics of programs: the Propositional Dynamic Logics (PDLs), which are modal logics specifically designed for reasoning about program schemes. The correspondence is based on the similarity between the interpretation structures of the two logics: at the extensional level, objects in DLs correspond to states in PDLs, whereas connections between two objects correspond to state transitions. At the intensional level, classes correspond to propositions, and roles corresponds to programs. The correspondence provides an invaluable tool for studying very expressive DLs. Indeed, it makes it clear that reasoning about assertions on classes is equivalent to reasoning about dynamic logic formulae (e.g., logical implication wrt a TBox, in any of the above logics, is equivalent to satisfiability of a specified dynamic logic formula), so that the large body of research on decision procedures for PDLs (see [16] for references) can be exploited in the setting of DLs.

However, in order to fully exploit this correspondence, at least three problems left open in [23] need to be solved, namely, how to fit functional restrictions on both atomic roles and their inverse, number restrictions, and assertions on individuals, respectively, into the correspondence. Note that these problems refer to points (1) and (3) above.

The work we have been carrying out on this subject [9; 10] has the explicit goal of providing suitable solutions to the above problems. Regarding point (1), we have investigated the following DL, named $\mathcal{CIF}$:

$$\begin{array}{lcl} C & ::= & A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \forall R.C \mid \exists R.C \mid \\ & & (\leq\ 1\ P) \mid (\leq\ 1\ P^-) \\ R & ::= & P \mid R_1 \sqcup R_2 \mid R_1 \circ R_2 \mid R^* \mid id(C) \mid R^- \end{array}$$

where $A$ and $P$ denote the generic atomic concept and role respectively. The main feature of $\mathcal{CIF}$ is the presence of functional restrictions on both atomic roles and their inverse. The decidability of the corresponding PDL, named $\mathcal{DIF}$, was not known. We have proved that satisfiability in $\mathcal{DIF}$ and logical implication for $\mathcal{CIF}$-TBoxes are EXPTIME-complete problems. The above decidability/complexity result holds also for $\mathcal{CIN}$ ($\mathcal{DIN}$), obtained from $\mathcal{CIF}$ ($\mathcal{DIF}$) by including the constructs for qualified number restrictions on both the atomic roles and their inverse. Moreover it is possible to polynomially encode n-ary relations among concepts in such logics. With respect to point (3), we have proved that for knowledge bases (TBox and ABox) expressed in two sublanguages of $\mathcal{CIF}$, namely $\mathcal{CI}$ (no functional restrictions) and $\mathcal{CF}$ (no inverse roles), satisfiability and logical implication are EXPTIME-complete. It is worth noting that, from the PDLs' point of view, an ABox has a natural counterpart: it can be regarded as a specification of partial computations.

Recently, both Schild and ourselves [24; 8] have pointed out that the correspondence between DLs and PDLs, can be extended to another logic of programs called (modal/propositional) mu-calculus [15] (see [26] for more references). This logic has the salient property of including explicit constructs for least and greatest fixpoints of formulae, which makes it more expressive than comparable PDLs. Indeed, the presence of the fixpoint constructs enables the logic to fully express inductive and co-inductive definitions, as well as to model, in a sin-

[3]In fact to correctly represent these data structures, the logic must also include constructs for inverse roles and for functional (or number) restrictions on both atomic role and their inverse.

gle framework, terminological cycles interpreted according to Least Fixpoint Semantics, Greatest Fixpoint Semantics, and Descriptive Semantics (see [21]). We have studied an extension of mu-calculus that includes qualified number restrictions on atomic roles, showing that satisfiability is EXPTIME-complete for it. Currently we are developing a method to reason with knowledge bases (ABox and TBox) expressed in a DL corresponding to mu-calculus extended with functional restrictions on atomic roles. We conclude remarking that a mu-calculus with a construct corresponding to inverse roles, though of great interest, has not been studied yet.

## References


[1] Giuseppe Attardi and Maria Simi. Consistency and completeness of omega, a logic for knowledge representation. In *Proc. of the Int. Joint Conf. on Artificial Intelligence IJCAI-81*, pages 504–510, 1981.

[2] Franz Baader and Bernhard Hollunder. A terminological knowledge representation system with complete inference algorithm. In *Proc. of the Workshop on Processing Declarative Knowledge, PDK-91*, Lecture Notes in Artificial Intelligence, pages 67–86. Springer-Verlag, 1991.

[3] Alexander Borgida and Peter F. Patel-Schneider. A semantics and complete algorithm for subsumption in the CLASSIC description logic. Submitted for publication, 1993.

[4] Ronald J. Brachman and Hector J. Levesque. The tractability of subsumption in frame-based description languages. In *Proc. of the 4th Nat. Conf. on Artificial Intelligence AAAI-84*, pages 34–37, 1984.

[5] Ronald J. Brachman, Deborah L. McGuinness, Peter F. Patel-Schneider, Lori Alperin Resnick, and Alex Borgida. Living with CLASSIC: when and how to use a KL-ONE-like language. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 401–456. Morgan Kaufmann, 1991.

[6] Ronald J. Brachman and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[7] Martin Buchheit, Francesco M. Donini, and Andrea Schaerf. Decidable reasoning in terminological knowledge representation systems. In *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence IJCAI-93*, pages 704–709, 1993.

[8] Giuseppe De Giacomo and Maurizio Lenzerini. Concept language with number restrictions and fixpoints, and its relationship with mu-calculus. To appear in ECAI-94.

[9] Giuseppe De Giacomo and Maurizio Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics (extended abstract). To appear in AAAI-94.

[10] Giuseppe De Giacomo and Maurizio Lenzerini. Description logics with inverse roles, functional restrictions, and n-ary relations. To appear in JELIA-94.

[11] Francesco M. Donini, Bernhard Hollunder, Maurizio Lenzerini, Alberto Marchetti Spaccamela, Daniele Nardi, and Werner Nutt. The complexity of existential quantification in concept languages. *Artificial Intelligence*, 2–3:309–327, 1992.

[12] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. The complexity of concept languages. In *Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning KR-91*, pages 151–162, 1991.

[13] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. Tractable concept languages. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence IJCAI-91*, pages 458–463, 1991.

[14] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, and Andrea Schaerf. Adding epistemic operators to concept languages. In *Proc. of the 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning KR-92*, pages 342–353, 1992.

[15] Dexter Kozen. Results on the propositional $\mu$-calculus. *Theoretical Computer Science* 27, pages 333–354, 1983.

[16] Dexter Kozen and Jerzy Tiuryn. Logics of programs. In *Handbook of Theoretical Computer Science – Formal Models and Semantics*, pages 789–840, Elsevier, 1990.

[17] Hector J. Levesque and Ron J. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3:78–93, 1987.

[18] Robert MacGregor. Inside the LOOM description classifier. *SIGART Bulletin*, 2(3):88–92, 1991.

[19] Bernhard Nebel. Computational complexity of terminological reasoning in BACK. *Artificial Intelligence*, 34(3):371–383, 1988.

[20] Bernhard Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43:235–249, 1990.

[21] Bernhard Nebel. Terminological cycles: Semantics and computational properties. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 331–361. Morgan Kaufmann, 1991.

[22] Peter F. Patel-Schneider. A hybrid, decidable, logic-based knowledge representation system. *Computational Intelligence*, 3(2):64–77, 1987.

[23] Klaus Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence IJCAI-91*, pages 466–471, 1991.

[24] Klaus Schild. Terminological cycles and the propositional $\mu$-calculus. In *Proc. of the 4th Int. Conf. on Principles of Knowledge Representation and Reasoning KR-94*. To appear, 1994.

[25] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[26] Colin Stirling. Modal and temporal logics. In *Handbook of Logic in Computer Science*, Vol. 2, pages 477–563, Clarendon Press, Oxford, 1992.