A Resolution Procedure for Description Logics with Nominal Schemas

Cong Wang and Pascal Hitzler

Kno.e.sis Center, Wright State University, Dayton OH 45435, USA
{cong,pascal}@knoesis.org

Abstract. We present a polynomial resolution-based decision procedure for the recently introduced description logic $\mathcal{ELHOV}_n(\Pi)$, which features nominal schemas as new language construct. Our algorithm is based on ordered resolution and positive superposition, together with a lifting lemma. In contrast to previous work on resolution for description logics, we have to overcome the fact that $\mathcal{ELHOV}_n(\Pi)$ does not allow for a normalization resulting in clauses of globally limited size.

1 Introduction

Description Logic (DL) and rule-based formalism are two prominent paradigms for Knowledge Representation. Although both paradigms are based in classical logic, they provide different expressivity and neither of them contains the other one, i.e. there exist axioms in DL which are not expressible in the rules paradigm and viceversa. Despite significant research efforts [4,5,17], many integrations of the two paradigms lead to undecidability (see [14] for a survey).

Currently, the most notable language in DLs family is the W3C recommendation Web Ontology Language (OWL). OWL can express many rules (see [14]), but it cannot express many others, such as

$$\exists \text{hasParent}(x, z) \land \exists \text{hasParent}(x, y) \land \text{married}(y, z) \rightarrow C(x) \quad (1)$$

which defines a class $C$ of children whose parents are married.

One idea for retaining decidability is to restrict the applicability of rules to named individuals. Rules that are understood in this sense are called $DL$-safe rules, and the combination between OWL DL and DL-safe rules is decidable [21].

Very recently, this idea is found to be able to carry further to description logic paradigm. Nominal schemas, as a new element of description logic syntax construct, was introduced in this sense [16]. It does not only further generalize the notion of DL-safety, but also enables to express the DL-safe rules within the description logic syntax. Using nominal schemas, rule (1) could be represented as:

$$\exists \text{hasParent}.\{z\} \sqcap \exists \text{hasParent}.\exists \text{married}.\{z\} \subseteq C \quad (2)$$

1 http://www.w3.org/TR/owl2-overview/
The expression \{z\} in (2) is a nominal schema, which is a variable that only binds with known individuals in a knowledge base and the binding is the same for all occurrences of the same nominal schema in an axiom.

Consequently, a new description logic \(\mathcal{SROIQV}_n\) was introduced, which indeed is a extension of OWL 2 DL \(\mathcal{SROIQ}\) with nominal schemas \(V_n\). It is decidable and has the same worst-case complexity as \(\mathcal{SROIQ}\) \[16\]. \(\mathcal{SROELV}_n\) is a tractable fragment of \(\mathcal{SROIQ}\). And extending \(\mathcal{SROELV}_n\) with role conjunction on simple roles and concept product would not change its worst-case complexity \[21\]. The importance of \(\mathcal{SROELV}_n(\cap_s, \times)\) is that it can incorporate OWL EL and OWL RL (two tractable fragments of OWL 2), and to allow restricted semantic interaction between the two. Also, it is more easy for ontology modelers to write rules in OWL syntax.

Although reasoning for description logic with nominal schema is theoretically feasible, the simple experiment in \[19\] shows that the naive approach, based on full grounding nominal schemas to all known individuals, is extremely slow. Therefore, it is really necessary to design a *smarter* algorithm. One idea is to ground nominal schemas in a more intelligent way, e.g. intelligent grounding, which is quite well known in Answer Set Programming (ASP) field \[23\]. In \[13\], the authors applied this strategy on \(\mathcal{ALCO}\) with nominal schema, but it needs a very good heuristics for grounding choices.

Another idea is to find a procedure that could do reasoning without grounding. We apply this idea in the paper by using resolution procedure with a lifting lemma. In this paper, we restrict \(\mathcal{SROELV}_n(\cap_s, \times)\) to \(\mathcal{ELHOV}_n(\cap)\), which disallows self role, complex role inclusion (role chain) and concept product, but allows role conjunction even for complex roles. The reason of this restriction will be discussed in Section 6. We provide a tractable resolution procedure for \(\mathcal{ELHO}(\cap)\), then show that the algorithm can also apply for \(\mathcal{ELHOV}_n(\cap)\) via a lifting lemma.

The structure of this paper is as follows. Section 2 describes some preliminaries of description logic and resolution procedure. Section 3 presents a tractable, sound and complete resolution procedure for \(\mathcal{ELHO}(\cap)\). Section 4 extends the algorithm to deal with \(\mathcal{ELHOV}_n(\cap)\). We will provide an example to illustrate the resolution procedure in Section 5 and then briefly discuss some related works in Section 6. Finally we conclude.

2 Preliminaries

2.1 Description Logic

We start by introducing the description logic \(\mathcal{ELHOV}_n(\cap)\).

A signature of \(\mathcal{ELHOV}_n(\cap)\) is a tuple \(\Sigma = \langle N_I, N_C, N_R, N_V \rangle\) of mutually disjoint finite sets of individual names, concept names, role names, and variables.