Neurocomputing 146 (2014) 249–263

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

ExpTime tableaux with global state caching for the description logic SHIO

Linh Anh Nguyen [a,b,*]

[a] Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
[b] Faculty of Information Technology, VNU University of Engineering and Technology, 144 Xuan Thuy, Hanoi, Vietnam

ARTICLE INFO

Article history:
Received 1 November 2013
Received in revised form
1 May 2014
Accepted 19 May 2014
Available online 26 June 2014

Keywords:
Description logics
Automated reasoning
Tableaux
Global state caching

ABSTRACT

We give the first ExpTime (complexity-optimal) tableau decision procedure for checking satisfiability of a knowledge base in the description logic $\mathcal{SHIO}$, which extends the basic description logic $\mathcal{ALC}$ with transitive roles, hierarchies of roles, inverse roles and nominals. Our procedure exploits global state caching and does not use blind (analytic) cuts.

1. Introduction

Collective intelligence refers to intelligence emerging from cooperation or competition of groups of (intelligent) agents. Achieving complex tasks by software agents requires cooperation supported by communication in a common language and conceptual space. Shared understanding of concepts and relationships is usually achieved by terminological knowledge and ontologies. Description logics (DLs) are formal languages for representing terminological knowledge. They provide a logical formalism for ontologies and the Semantic Web.

Automated reasoning in DLs is useful in engineering and querying ontologies. One of the basic reasoning problems in DLs is to check satisfiability of a knowledge base in a considered DL. Most of other reasoning problems in DLs are reducible to this one. For example, in the Semantic Web and multiagent applications, ontology fusion frequently leads to inconsistencies, and paraconsistent reasoning with inconsistent DL-based ontologies can be reduced to the mentioned satisfiability problem (in a traditional two-valued DL) [18,28].

DLs represent the domain of interest in terms of concepts, individuals, and roles. A concept is interpreted as a set of individuals, while a role is interpreted as a binary relation among individuals. A knowledge base in a DL consists of axioms about roles (grouped into an RBox), terminology axioms (grouped into a TBox), and assertions about individuals (grouped into an ABox). A DL is usually specified by (i) a set of constructors that allow building complex concepts and complex roles from concept names, role names and individual names, (ii) allowed forms of axioms and assertions. The basic DL $\mathcal{ALC}$ allows basic concept constructors listed in Table 1, but does not allow role constructors or role axioms. The most common additional features for extending $\mathcal{ALC}$ are also listed in Table 1 together with syntax and examples: $\mathcal{I}$ is a role constructor, $\mathcal{Q}$ and $\mathcal{O}$ are concept constructors, while $\mathcal{H}$ and $\mathcal{S}$ are allowed forms of role axioms. The name of a DL is usually formed by the names of its additional features, as in the cases of $\mathcal{SH}$, $\mathcal{SHI}$, $\mathcal{SHIQ}$, $\mathcal{SHIO}$, $\mathcal{SHOQ}$ and $\mathcal{SHOIQ}$. $\mathcal{SROIQ}$ [10] is a further expressive DL used as the logical base for the Web Ontology Language OWL 2 DL. The satisfiability problem is ExpTime-complete in $\mathcal{ALC}$, $\mathcal{SH}$, $\mathcal{SHI}$, $\mathcal{SHIQ}$, $\mathcal{SHIO}$ and $\mathcal{SHOQ}$, NExpTime-complete in $\mathcal{SHOIQ}$, and N2ExpTime-complete in $\mathcal{SROIQ}$. Notice the jump from ExpTime-complete to NExpTime-complete when $\mathcal{I}$, $\mathcal{Q}$, $\mathcal{O}$ interact with each others.

In this paper we study automated reasoning in the DL $\mathcal{SHIO}$.

1.1. Related work and motivations

In [8] Hladik and Model described tableau systems for the problem of checking satisfiability of a concept w.r.t. an RBox in the DLs $\mathcal{SHIO}$ and $\mathcal{SHIQ}$. This problem is less general than the problem of checking satisfiability of a knowledge base. (In $\mathcal{SHIO}$ and $\mathcal{SHIQ}$, TBox axioms can be "internalized"; and in $\mathcal{SHIO}$, by using nominals, ABox assertions can be simulated by concepts.

* Correspondence address: Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland.
E-mail address: nguyen@mimuw.edu.pl

http://dx.doi.org/10.1016/j.neucom.2014.05.071
0925-2312/

Table 1
Concept constructors for $\mathcal{ALC}$ and some additional constructors/features of other DLs.

Constructor/feature	Syntax	Example
Complement	$\neg C$	$\neg Male$
Intersection	$C \sqcap D$	$Human \sqcap Male$
Union	$C \sqcup D$	$Doctor \sqcup Lawyer$
Existential restriction	$\exists r.C$	$\exists hasChild.Male$
Universal restriction	$\forall r.C$	$\forall hasChild.Female$
Inverse roles ($\mathcal{I}$)	r^-	$hasChild^-$ (i.e., $hasParent$)
Quantified number	$\geq n\,R.C$	$\geq 3\ hasChild.Male$
restrictions ($\mathcal{Q}$)	$\leq n\,R.C$	$\leq 2\ hasParent.\top$
Nominals ($\mathcal{O}$)	$\{a\}$	$\{John\}$
Hierarchies of roles ($\mathcal{H}$)	$R \sqsubseteq S$	$hasChild \sqsubseteq hasDesc$
Transitive roles ($\mathcal{S}$)	$R \circ R \sqsubseteq R$	$hasDesc \circ hasDesc \sqsubseteq hasDesc$

However, the approach with such reductions is less efficient.) Using the given tableau systems, the authors claimed that they could derive automata algorithms deciding satisfiability of $\mathcal{SHIO}/\mathcal{SHIQ}$ concepts w.r.t. RBoxes in ExpTime and terminating tableau algorithms. They also stated that "*In short, the advantages of automata algorithms are on the theoretical side, because in many cases the proofs are very elegant and provide tight complexity bounds* (*in particular for* ExpTime*-complete logics*), *whereas the advantages of tableau algorithms are on the practical side, since they are well suited for implementation and optimization*." Note that the tableau algorithms given in [8] have only the termination property but not the ExpTime complexity. In particular, the tableau algorithm given of [8] for $\mathcal{SHIO}$ runs in NExpTime.

In [12] Horrocks and Sattler gave a tableau decision procedure for $\mathcal{SHOIQ}$, and in [10] Horrocks et al. gave a tableau decision procedure for $\mathcal{SROIQ}$. As $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$ are more general than $\mathcal{SHIO}$, these decision procedures can be used for reasoning in $\mathcal{SHIO}$. However, both of them have a non-optimal complexity,[1] and even when restricted to checking consistency of a concept with respect to an RBox and a TBox in $\mathcal{SHI}$ as in [11], they still have a NExpTime complexity. One of the reasons is that these decision procedures use backtracking to deal with disjunction $\sqcup$ and "or"-branching (e.g., the "choice"-rule).

By applying global caching [33,4], together with colleagues (Goré, Szałas and Dunin-Kęplicz) we have developed complexity-optimal (ExpTime) tableau decision procedures for a number of modal and description logics with ExpTime complexity (see [4,21] for references). In [3,25,27,1] blind (analytic) cuts are used to deal with inverse roles and converse modal operators. As blind cuts may be not efficient in practice, Goré and Widmann developed the first blind-cuts-free ExpTime tableau decision procedures, based on global state caching, for the DL $\mathcal{ALCI}$ (for the case without ABoxes) [5] and CPDL [6]. (Their procedures still use a kind of cuts, let us call them *cuts-on-demand*, which are much better than blind cuts.) We have applied global state caching to the modal logic CPDL_{reg} [22] and the DLs $\mathcal{ALCI}$ [20], $\mathcal{SHI}$ [19], $\mathcal{SHIQ}$ [21,23] for the case with ABoxes to obtain blind-cuts-free ExpTime tableau decision procedures for them.

Dealing with nominals and qualified number restrictions in developing complexity-optimal tableaux requires advanced techniques. In the paper [5] on $\mathcal{ALCI}$, Goré and Widmann wrote "*The extension to role hierarchies and transitive roles should not present difficulties, but the extension to include nominals and qualified number restrictions is not obvious to us.*" In [21,23], by using global state caching and integer linear feasibility checking, we have been succeeded in developing the first ExpTime (complexity-optimal) tableau decision procedure for checking satisfiability of a knowledge base in the DL $\mathcal{SHIQ}$. Recently, together with Golińska-Pilarek, we have also given the first ExpTime tableau decision procedure for the DL $\mathcal{SHOQ}$ [24], which is based on global caching.

[1] That is, higher than NExpTime for $\mathcal{SHOIQ}$ and higher than N2ExpTime for $\mathcal{SROIQ}$.

1.2. Our contributions

Among the three well-known expressive DLs $\mathcal{SHIQ}$, $\mathcal{SHOQ}$, $\mathcal{SHIO}$ with ExpTime complexity, it remains to develop ExpTime tableaux for $\mathcal{SHIO}$. (Recall that the more expressive DL $\mathcal{SHOIQ}$ is NExpTime-complete.)

In this paper we present the first tableau decision procedure with an ExpTime (optimal) complexity for checking satisfiability of a knowledge base in the DL $\mathcal{SHIO}$. Our procedure exploits global state caching and does not use blind (analytic) cuts.

1.3. The structure of this paper

The rest of this paper is structured as follows. In Section 2 we recall the notation and semantics of $\mathcal{SHIO}$. In Section 3 we present our tableau decision procedure for $\mathcal{SHIO}$. We start with the data structure and the framework of our procedure. We then give a few illustrative examples. After that we specify and explain the used tableau rules. We finish the section by stating theoretical results about our procedure. We conclude the paper in Section 4.

2. Notation and semantics of $\mathcal{SHIO}$

Our language uses a countable set $\mathbf{C}$ of *concept names*, a countable set $\mathbf{R}$ of *role names*, and a countable set $\mathbf{I}$ of *individual names*. A concept name stands for a unary predicate, a role name stands for a binary predicate, and an individual name stands for a constant. We use letters like A and B for concept names, r and s for role names, and a and b for individual names. We refer to A and B also as *atomic concepts*, and to a and b as *individuals*.

For $r \in \mathbf{R}$, let r^- be a new symbol, called the *inverse* of r. Let $\mathbf{R}^- = \{r^- \mid r \in \mathbf{R}\}$ be the set of *inverse roles*. For $r \in \mathbf{R}$, define $(r^-)^- = r$. A *role* is any member of $\mathbf{R} \cup \mathbf{R}^-$. We use letters like R and S to denote roles.

An ($\mathcal{SHIO}$) *RBox* $\mathcal{R}$ is a finite set of role axioms of the form $R \sqsubseteq S$ or $R \circ R \sqsubseteq R$. By $ext(\mathcal{R})$ we denote the least extension of $\mathcal{R}$ such that

- $R \sqsubseteq R \in ext(\mathcal{R})$ for any role R
- if $R \sqsubseteq S \in ext(\mathcal{R})$ then $R^- \sqsubseteq S^- \in ext(\mathcal{R})$
- if $R \circ R \sqsubseteq R \in ext(\mathcal{R})$ then $R^- \circ R^- \sqsubseteq R^- \in ext(\mathcal{R})$
- if $R \sqsubseteq S \in ext(\mathcal{R})$ and $S \sqsubseteq T \in ext(\mathcal{R})$ then $R \sqsubseteq T \in ext(\mathcal{R})$.

By $R \sqsubseteq_{\mathcal{R}} S$ we mean $R \sqsubseteq S \in ext(\mathcal{R})$, and by $trans_{\mathcal{R}}(R)$ we mean $(R \circ R \sqsubseteq R) \in ext(\mathcal{R})$. If $R \sqsubseteq_{\mathcal{R}} S$ then R is a *subrole* of S (w.r.t. $\mathcal{R}$). If $trans_{\mathcal{R}}(R)$ then R is a *transitive role* (w.r.t. $\mathcal{R}$).

Concepts in $\mathcal{SHIO}$ are formed using the following BNF grammar:

$$C, D ::= \top \mid \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C \mid \{a\}$$

A concept stands for a set of individuals. The concept $\top$ stands for the set of all individuals (in the considered domain). The concept $\bot$ stands for the empty set. We use letters like C and D to denote arbitrary concepts.

A *TBox* is a finite set of axioms of the form $C \sqsubseteq D$ or $C \doteq D$.

An *ABox* is a finite set of *assertions* of the form $a : C$, $R(a,b)$ or $a \not\doteq b$.

An axiom $C \sqsubseteq D$ means C is a subconcept of D, while $C \doteq D$ means C and D are equivalent concepts. An assertion $a : C$ means a