Efficient Reasoning with Range and Domain Constraints

Dmitry Tsarkov and Ian Horrocks

Department of Computer Science
The University of Manchester
Manchester, UK
{tsarkov|horrocks}@cs.man.ac.uk

Abstract

We show how a tableaux algorithm for $\mathcal{SHIQ}$ can be extended to support role boxes that include range and domain axioms, prove that the extended algorithm is still a decision procedure for the satisfiability and subsumption of $\mathcal{SHIQ}$ concepts w.r.t. such a role box, and show how support for range and domian axioms can be exploited in order to add a new form of absorption optimisation called role absorption. We illustrate the effectiveness of the optimised algorithm by analysing the perfomance of our FaCT++ implementation when classifying terminologies derived from realistic ontologies.

1 Introduction

Many modern ontology languages (e.g., OIL [3], DAML+OIL [8] and OWL [2]) are based on expressive description logics, and in particular on the $\mathcal{SHIQ}$ family of description logics [9]. These ontology languages typically support domain and range constraints on roles, i.e., axioms asserting that if an individual x is related to an individual y by a role R, then x must be an instance of the concept that is the domain of R and y must be an instance of the concept that is the range of R.Such axioms are not directly supported by $\mathcal{SHIQ}$, but can trivially be transformed into *general inclusion axioms* (GCIs), i.e., an axiom asserting a subsumption relationship between two arbitrary concept terms. In particular, restricting the domain of a role R to be concept C is equivalent to adding an axiom $\exists R.\top \sqsubseteq C$, and restricting the range of a role R to be concept D is equivalent to adding an axiom $\top \sqsubseteq \forall R.D$.

The problem with this transformation is that such GCIs are not amenable to *absorption*, an optimisation technique that tries to rewrite GCIs so that they can be efficiently dealt with using the *lazy unfolding* optimisation [6]. Absorption is one of the crucial optimisations that enable state of the art DL reasoners such as FaCT [7], Racer [5] and Pellet [12] to deal effectively with large knowledge bases (KBs), and these reasoners perform much less well with KBs containing significant numbers of unabsorbable GCIs. Unfortunately, many ontologies contain large numbers of different roles, each with a range and domain constraint, and the resulting KBs therefore contain many unabsorbable GCIs.

It has already been shown that, in order for the Racer system to be able to classify large KBs containing many range and domain constraints, it is necessary to give a special treatment to the GCIs introduced by range and domain axioms [4]. The approach used by Racer is to extend the lazy unfolding optimisation so that concepts equivalent to those that would be introduced by the GCIs are introduced only as necessary. In the approach presented here, we extend the tableaux satisfiability testing algorithm so that range and domain axioms are directly supported. The advantage with this approach is that we are able to extend the formal correctness proof to demonstrate that the extended algorithm is still a decision procedure for $\mathcal{SHIQ}$ satisfiability (i.e., it returns *satisfiable* iff the input concept is satisfiable).

As well as allowing range and domain to be dealt with very efficiently, this algorithm also allows us to implement an extended version of the absorption optimisation, called *role absorption*, that transforms GCIs into domain constraints. Role absorption can provide alternative and perhaps more effective ways to absorb certain forms of GCI, and can also be applied to some otherwise unabsorbable forms of GCI. This can lead to dramatic performance improvements for KBs that contain significant numbers of such GCIs. We demonstrate this (as well as the performance improvements resulting from support for range and domain axioms) with an empirical analysis of the performance of the extended algorithm when classifying several KBs derived from realistic ontologies.

2 Preliminaries

We first introduce the syntax and semantics of the $\mathcal{SHIQ}$ logic, including the semantics of role boxes extended with range and domain axioms. Most details of the logic and the tableaux algorithm are little changed from those presented in [9]. We will, therefore, focus mainly on the parts that have been added in order to deal with range and domain axioms, and refer the reader to [9] for complete information on the remainder.

The absolutely most part of formal definitions here is taken from [9]. We have introduced new constructions into the existing definitions, so all algorithms were slightly changed.

Definition 1 *Let* $\mathbf{C}$ *and* $\mathbf{R}$ *be disjoint sets of* concept names *and* role names *respectively. The set of* $\mathcal{SHIQ}$-roles *is* $\mathbf{R} \cup \{R^- \mid R \in \mathbf{R}\}$. *To avoid considering roles such as* R^{--}, *we define a function* Inv *on roles such that* $\mathsf{Inv}(R) = R^-$ *if* R *is a role name, and* $\mathsf{Inv}(R) = S$ *if* $R = S^-$. *For* R *and* S $\mathcal{SHIQ}$*-roles and* C *a* $\mathcal{SHIQ}$*-concept, a* role axiom *is either a role inclusion of the form* $R \sqsubseteq S$, *a transitivity axiom of the form* $\mathsf{Trans}(R)$, *or a* constraint axiom *of the form* $\mathsf{Domain}(R, C)$ *or* $\mathsf{Range}(R, C)$. *A* role box $\mathcal{R}$ *is a finite set of role axioms.*

A role R *is called* simple *if, for* $\underline{\sqsubseteq}^*$ *the transitive reflexive closure of* $\sqsubseteq$ *on* $\mathcal{R}$ *and for each role* S, $S \underline{\sqsubseteq}^* R$ *implies* $\mathsf{Trans}(S) \notin \mathcal{R}$ *and* $\mathsf{Trans}(\mathsf{Inv}(S)) \notin \mathcal{R}$.

The set of concepts *is the smallest set such that every concept name is a concept, and, for* C *and* D *concepts,* R *a role,* S *a simple role and* n *a non-negative integer, then* $C \sqcap D$, $C \sqcup D$, $\neg C$, $\exists R.C$, $\forall R.C$, $\geqslant nS.C$ *and* $\leqslant nS.C$ *are also concepts.*

The semantics is given by means of an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *consisting of a non-empty set* $\Delta^{\mathcal{I}}$, *called the* domain *of* $\mathcal{I}$, *and a* valuation $\cdot^{\mathcal{I}}$ *which maps every concept to a subset of* $\Delta^{\mathcal{I}}$ *and every role to a subset of* $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ *such that, for all concepts* C, D, *roles* R, S, *and non-negative integers* n, *the properties in Figure 1 are satisfied, where* $\sharp M$ *denotes the cardinality of a set* M.

An interpretation satisfies a role axiom if it satisfies the semantic conditions given in Figure 1. An interpretation satisfies a role box $\mathcal{R}$ *if it satisfies each role axiom in* $\mathcal{R}$.

A terminology *or* TBox $\mathcal{T}$ *is a finite set of* general concept inclusion axioms, $\mathcal{T} = \{C_1 \sqsubseteq D_1, \ldots, C_n \sqsubseteq D_n\}$, *where* C_i, D_i *are arbitrary* $\mathcal{SHIQ}$*-concepts. An interpretation* $\mathcal{I}$ *satisfies* $\mathcal{T}$ *iff* $C_i^{\mathcal{I}} \subseteq D_i^{\mathcal{I}}$ *holds for all* $C_i \sqsubseteq D_i \in \mathcal{T}$.

Concepts & Roles	Syntax	Semantics
atomic concept $\mathbf{C}$	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
atomic role $\mathbf{R}$	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
inverse role	R^-	$\{\langle x, y\rangle \mid \langle y, x\rangle \in R^{\mathcal{I}}\}$
conjunction	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
negation	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
exists restriction	$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\}$
value restriction	$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}}$ implies $y \in C^{\mathcal{I}}\}$
atleast restriction	$\geqslant nS.C$	$(\geqslant nS.C)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in S^{\mathcal{I}}\} \cap C^{\mathcal{I}}) \geqslant n\}$
atmost restriction	$\leqslant nS.C$	$(\leqslant nS.C)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in S^{\mathcal{I}}\} \cap C^{\mathcal{I}}) \leqslant n\}$
Role Axioms	**Syntax**	**Semantics**
role inclusion	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
transitive role	$\mathsf{Trans}(R)$	$R^{\mathcal{I}} = (R^+)^{\mathcal{I}}$
role domain	$\mathsf{Domain}(R, C)$	$\langle x, y\rangle \in R^{\mathcal{I}}$ implies $x \in C^{\mathcal{I}}$
role range	$\mathsf{Range}(R, C)$	$\langle x, y\rangle \in R^{\mathcal{I}}$ implies $y \in C^{\mathcal{I}}$

Figure 1: Syntax and semantics of $\mathcal{SHIQ}$

A $\mathcal{SHIQ}$-concept C is satisfiable *w.r.t. a role box $\mathcal{R}$ and a terminology $\mathcal{T}$ iff there is an interpretation $\mathcal{I}$ with $C^{\mathcal{I}} \neq \emptyset$ that satisfies both $\mathcal{R}$ and $\mathcal{T}$. Such an interpretation is called a* model *of C w.r.t. $\mathcal{R}$ and $\mathcal{T}$. A concept C is* subsumed *by a concept D w.r.t. $\mathcal{R}$ and $\mathcal{T}$ iff $C^{\mathcal{I}} \sqsubseteq D^{\mathcal{I}}$ for each interpretation $\mathcal{I}$ satisfying $\mathcal{R}$ and $\mathcal{T}$.*

Theorem 1 *Satisfiability and subsumption of $\mathcal{SHIQ}$-concepts w.r.t. terminologies and role boxes is polynomially reducible to (un)satisfiability of $\mathcal{SHIQ}$-concepts w.r.t. role boxes [9].*

3 Tableaux Reasoning with Range and Domain

Here we present an algorithm for deciding the satisfiability of a $\mathcal{SHIQ}$-concept C w.r.t. a role box $\mathcal{R}$; it is an extension of the $\mathcal{SHIQ}$ tableaux algorithm from [9].

For ease of Tableaux construction, we assume C and all concepts in (range and domain axioms in) $\mathcal{R}$ to be in *negation normal form* (NNF), that is, negation occurs only in front of concept names. Any $\mathcal{SHIQ}$-concept can easily be transformed into an equivalent one in NNF by pushing negations inwards; with $\sim C$ we denote the NNF of $\neg C$. We define $\mathsf{RD}(\mathcal{R})$ as the set of concepts s.t. $C \in \mathsf{RD}(\mathcal{R})$ iff $\mathsf{Domain}(R, C) \in \mathcal{R}$ or $\mathsf{Range}(R, C) \in \mathcal{R}$ for some role R. We define $\mathtt{cl}(C, \mathcal{R})$ as the smallest set of concepts that is a superset of $C \cup \mathsf{RD}(\mathcal{R})$ and is closed under subconcepts and $\sim$.

Definition 2 *Let D be a $\mathcal{SHIQ}$-concept in NNF, $\mathcal{R}$ a role box, and $\mathbf{R}_D$ the set of roles occurring in D and $\mathcal{R}$ together with their inverses. Then $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ is a* tableau *for D w.r.t. $\mathcal{R}$ iff $\mathbf{S}$ is a set of individuals, $\mathcal{L} : \mathbf{S} \to 2^{\mathtt{cl}(D,\mathcal{R})}$ maps each individual to a set of concepts, $\mathcal{E} : \mathbf{R}_D \to 2^{\mathbf{S}\times\mathbf{S}}$ maps each role to a set of pairs of individuals, and there is some individual $s \in \mathbf{S}$ such that $D \in \mathcal{L}(s)$. Furthermore, for all $s, t \in \mathbf{S}$, $C, C_1, C_2 \in \mathtt{cl}(D, \mathcal{R})$, and $R, S \in \mathbf{R}_D$, it holds that:*

1. *if* $C \in \mathcal{L}(s)$*, then* $\neg C \notin \mathcal{L}(s)$*,*
2. *if* $C_1 \sqcap C_2 \in \mathcal{L}(s)$*, then* $C_1 \in \mathcal{L}(s)$ *and* $C_2 \in \mathcal{L}(s)$*,*
3. *if* $C_1 \sqcup C_2 \in \mathcal{L}(s)$*, then* $C_1 \in \mathcal{L}(s)$ *or* $C_2 \in \mathcal{L}(s)$*,*
4. *if* $\forall S.C \in \mathcal{L}(s)$ *and* $\langle s,t\rangle \in \mathcal{E}(S)$*, then* $C \in \mathcal{L}(t)$*,*
5. *if* $\exists S.C \in \mathcal{L}(s)$*, then there is some* $t \in \mathbf{S}$ *such that* $\langle s,t\rangle \in \mathcal{E}(S)$ *and* $C \in \mathcal{L}(t)$*,*
6. *if* $\forall S.C \in \mathcal{L}(s)$ *and* $\langle s,t\rangle \in \mathcal{E}(R)$ *for some* $R \mathrel{\underline{\circledast}} S$ *with* $\mathsf{Trans}(R)$*, then* $\forall R.C \in \mathcal{L}(t)$*,*
7. $\langle s,t\rangle \in \mathcal{E}(R)$ *iff* $\langle t,s\rangle \in \mathcal{E}(\mathsf{Inv}(R))$*,*
8. *if* $\langle s,t\rangle \in \mathcal{E}(R)$ *and* $R \mathrel{\underline{\circledast}} S$*, then* $\langle s,t\rangle \in \mathcal{E}(S)$*,*
9. *if* $(\leqslant n\ S\ C) \in \mathcal{L}(s)$*, then* $\sharp S^T(s,C) \leqslant n$*,*
10. *if* $(\geqslant n\ S\ C) \in \mathcal{L}(s)$*, then* $\sharp S^T(s,C) \geqslant n$*,*
11. *if* $(\bowtie\ n\ S\ C) \in \mathcal{L}(s)$ *and* $\langle s,t\rangle \in \mathcal{E}(S)$ *then* $C \in \mathcal{L}(t)$ *or* ${\sim}C \in \mathcal{L}(t)$*,*
12. *if* $\langle s,t\rangle \in \mathcal{E}(S)$ *and* $\mathsf{Domain}(S,C) \in \mathcal{R}$*, then* $C \in \mathcal{L}(s)$*,*
13. *if* $\langle s,t\rangle \in \mathcal{E}(S)$ *and* $\mathsf{Range}(S,C) \in \mathcal{R}$*, then* $C \in \mathcal{L}(t)$*,*

where we use $\bowtie$ *as a placeholder for both* $\leqslant$ *and* $\geqslant$ *and we define*

$$S^T(s,C) := \{t \in \mathbf{S} \mid \langle s,t\rangle \in \mathcal{E}(S) \text{ and } C \in \mathcal{L}(t)\}.$$

Lemma 1 *A* $\mathcal{SHIQ}$*-concept* D *is satisfiable w.r.t. a role box* $\mathcal{R}$ *iff* D *has a tableau w.r.t.* $\mathcal{R}$*.*

3.1 An Extended Tableaux Algorithm

In order to make the following description easier, we will abuse notation by using $\mathsf{Domain}(R)$ and $\mathsf{Range}(R)$ to mean the sets of concepts corresponding to the domain and range axioms in $\mathcal{R}$ that apply to a role R, i.e., $\mathsf{Domain}(R) = \{C \mid \mathsf{Domain}(R,C) \in \mathcal{R}\}$, and $\mathsf{Range}(R) = \{C \mid \mathsf{Range}(R,C) \in \mathcal{R}\}$.

Definition 3 *A* completion tree *for a concept* D *is a tree where each node* x *of the tree is labelled with a set* $\mathcal{L}(x) \subseteq \mathtt{cl}(D,\mathcal{R})$ *and each edge* $\langle x,y\rangle$ *is labelled with a set* $\mathcal{L}(\langle x,y\rangle)$ *of (possibly inverse) roles occurring in* $\mathtt{cl}(D,\mathcal{R})$*; explicit inequalities between nodes of the tree are recorded in a binary relation* $\not\doteq$ *that is implicitly assumed to be symmetric.*

Given a completion tree, a node y *is called an* R-successor *of a node* x *iff* y *is a successor of* x *and* $S \in \mathcal{L}(\langle x,y\rangle)$ *for some* S *with* $S \mathrel{\underline{\circledast}} R$*. A node* y *is called an* R-neighbour *of* x *iff* y *is an* R*-successor of* x*, or if* x *is an* $\mathsf{Inv}(R)$*-successor of* y*. Predecessors and ancestors are defined as usual.*

A node is blocked *iff it is directly or indirectly blocked. A node* x *is* directly blocked *iff none of its ancestors are blocked, and it has ancestors* x'*,* y *and* y' *such that*

1. x *is a successor of* x' *and* y *is a successor of* y' and
2. $\mathcal{L}(x) = \mathcal{L}(y)$ *and* $\mathcal{L}(x') = \mathcal{L}(y')$ and
3. $\mathcal{L}(\langle x',x\rangle) = \mathcal{L}(\langle y',y\rangle)$*.*

A node y is indirectly blocked *iff one of its ancestors is blocked, or it is a successor of a node x and $\mathcal{L}(\langle x, y \rangle) = \emptyset$.*

For a node x, $\mathcal{L}(x)$ is said to contain a clash *iff $\{A, \neg A\} \subseteq \mathcal{L}(x)$ or if, for some concept C, some role S, and some $n \in \mathbb{N}$: $(\leqslant n\ S\ C) \in \mathcal{L}(x)$ and there are $n+1$ S-neighbours $y_0, \ldots, y_n$ of x such that $C \in \mathcal{L}(y_i)$ and $y_i \not\doteq y_j$ for all $0 \leq i < j \leq n$. A completion tree is called* clash-free *iff none of its nodes contains a clash; it is called* complete *iff none of the expansion rules is applicable.*

For a $\mathcal{SHIQ}$-concept D, the algorithm starts with a completion tree consisting of a single node x with $\mathcal{L}(x) = \{D\}$ and $\not\doteq = \emptyset$. It applies the expansion rules in Fig. 2, stopping when a clash occurs, and answers "D is satisfiable" iff the completion rules can be applied in such a way that they yield a complete and clash-free completion tree.

Note that the only change w.r.t. [9] is addition of the *domain* and *range*-rules that add concepts to node labels as required by domain and range axioms.

Lemma 2 *Let D be an $\mathcal{SHIQ}$-concept.*

1. *The tableaux algorithm terminates when started with D.*
2. *If the expansion rules can be applied to D such that they yield a complete and clash-free completion tree, then D has a tableau.*
3. *If D has a tableau, then the expansion rules can be applied to D such that they yield a complete and clash-free completion tree.*

The following theorem is an immediate consequence of Lemmas 1, 2 and Theorem 1.

Theorem 2 *The tableaux algorithm is a decision procedure for the satisfiability and subsumption of $\mathcal{SHIQ}$-concepts with respect to role boxes.*

4 Role Absorption

Given that the new algorithm is able to deal directly with range and domain axioms, it makes sense to transform GCIs into range and domain axioms. We call this new form of absorption *role absorption* in contrast to the usual form of absorption we will refer to as *concept absorption* (see [10]).

Role absorption is important because in ontology derived KBs range and domain constraints will often have been transformed into GCIs. This is because tools such as OilEd [1] and Protégé [11] are designed to work with range of DL reasoners, some of which (e.g., FaCT) do not support range and domain axioms. Moreover, these forms of GCI are not, in general, amenable to standard concept absorption techniques.

We introduce two kinds of role absorption: *basic* and *extended* role absorptions.

Basic role absorption.

The simple form of role absorption, which we will refer to as *basic role absorption*, deals with the axiom of the form $\exists R.\top \sqsubseteq C$ and $\top \sqsubseteq \forall R.C$ and is formalised in the following theorem:

$\sqcap$-rule:	if 1. $C_1 \sqcap C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. $\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$ then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{C_1, C_2\}$
$\sqcup$-rule:	if 1. $C_1 \sqcup C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. $\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$ then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{C\}$ for some $C \in \{C_1, C_2\}$
$\exists$-rule:	if 1. $\exists S.C \in \mathcal{L}(x)$, x is not blocked, and 2. x has no S-neighbour y with $C \in \mathcal{L}(y)$, then create a new node y with $\mathcal{L}(\langle x, y\rangle) = \{S\}$ and $\mathcal{L}(y) = \{C\}$
$\forall$-rule:	if 1. $\forall S.C \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. there is an S-neighbour y of x with $C \notin \mathcal{L}(y)$ then $\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{C\}$
$\forall_+$-rule:	if 1. $\forall S.C \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. there is some R with $\mathsf{Trans}(R)$ and $R \sqsubseteq\!\!\!*\, S$, 3. there is an R-neighbour y of x with $\forall R.C \notin \mathcal{L}(y)$ then $\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{\forall R.C\}$
choose-rule:	if 1. $(\bowtie\ n\ S\ C) \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. there is an S-neighbour y of x with $\{C, \sim C\} \cap \mathcal{L}(y) = \emptyset$ then $\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{E\}$ for some $E \in \{C, \sim C\}$
$\geqslant$-rule:	if 1. $(\geqslant n\ S\ C) \in \mathcal{L}(x)$, x is not blocked, and 2. there are not n S-neighbours $y_1, \ldots, y_n$ of x with $C \in \mathcal{L}(y_i)$ and $y_i \neq y_j$ for $1 \leq i < j \leq n$ then create n new nodes $y_1, \ldots, y_n$ with $\mathcal{L}(\langle x, y_i\rangle) = \{S\}$, $\mathcal{L}(y_i) = \{C\}$, and $y_i \neq y_j$ for $1 \leq i < j \leq n$.
$\leqslant$-rule:	if 1. $(\leqslant n\ S\ C) \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. $\sharp S^{\mathbf{T}}(x, C) > n$ and there are two S-neighbours y, z of x with $C \in \mathcal{L}(y), C \in \mathcal{L}(z)$, y is not an ancestor of x, and not $y \neq z$ then 1. $\mathcal{L}(z) \longrightarrow \mathcal{L}(z) \cup \mathcal{L}(y)$ and 2. if z is an ancestor of x then $\mathcal{L}(\langle z, x\rangle) \longrightarrow \mathcal{L}(\langle z, x\rangle) \cup \mathsf{Inv}(\mathcal{L}(\langle x, y\rangle))$ else $\mathcal{L}(\langle x, z\rangle) \longrightarrow \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}(\langle x, y\rangle)$ 3. $\mathcal{L}(\langle x, y\rangle) \longrightarrow \emptyset$ 4. Set $u \neq z$ for all u with $u \neq y$
domain-rule	if 1. $C \in \mathsf{Domain}(S)$, x is not indirectly blocked, and 2. there is an S-neighbour y of x and $C \notin \mathcal{L}(x)$ then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{C\}$
range-rule	if 1. $C \in \mathsf{Range}(S)$, x is not indirectly blocked, and 2. there is an S-neighbour y of x with $C \notin \mathcal{L}(y)$ then $\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{C\}$

Figure 2: The complete tableaux expansion rules for $\mathcal{SHIQ}$

Theorem 3 *Let $\mathcal{R}$ be a $\mathcal{SHIQ}$ role box.*

1. *An interpretation $\mathcal{I}$ satisfies $\mathcal{R}$ and $\exists R.\top \sqsubseteq C$ iff $\mathcal{I}$ satisfies $\mathcal{R} \cup \{\mathsf{Domain}(R, C)\}$.*
2. *An interpretation $\mathcal{I}$ satisfies $\mathcal{R}$ and $\top \sqsubseteq \forall R.C$ iff $\mathcal{I}$ satisfies $\mathcal{R} \cup \{\mathsf{Range}(R, C)\}$.*

Extended Role Absorption

Rewriting techniques similar to those used in concept absorption can be used to extend the basic role absorption technique to deal with a wider range of axioms. An axiom of the form $\exists R.C \sqsubseteq D$ can be absorbed into a domain constraint $\mathsf{Domain}(R, D \sqcup \neg\exists R.C)$ by rewriting it as $\exists R.\top \sqsubseteq D \sqcup \neg\exists R.C$. Similarly, an axiom of the form $D \sqsubseteq \forall R.C$ can be absorbed into a domain constraint $\mathsf{Domain}(R, \neg D \sqcup \neg\exists R.\neg C)$.

5 Implementation and Empirical Evaluation

We have implemented the extended tableaux algorithm and role absorption optimisation in the FaCT++ DL reasoner. FaCT++ is a next generation of the well-known FaCT reasoner [7], being developed as part of the EU WonderWeb project (see `http://wonderweb.semanticweb.org/`); it is based on the same tableaux algorithms as the original FaCT, but has a different architecture and is written in C++ instead of Lisp.

Absorption

Absorption in FaCT++ uses the same basic approach as FaCT [10, 6]. Given a TBox $\mathcal{T}$, the absorption algorithm constructs a triple of TBoxes $\langle \mathcal{T}_{\mathrm{def}}, \mathcal{T}_{\mathrm{sub}}, \mathcal{T}_{\mathrm{g}} \rangle$ such that:

- $\mathcal{T}_{\mathrm{def}}$ is a set of axioms of the form $A \equiv C$ (equivalent to a pair of axioms $\{A \sqsubseteq C, C \sqsubseteq A\} \subseteq \mathcal{T}$), where $A \in \mathbf{C}$ (i.e., A is a concept name) and there is most one such axiom for each $A \in \mathbf{C}$. Such an axiom is often called a *definition* (of A).
- $\mathcal{T}_{\mathrm{sub}}$ consists of a set of axioms of the form $A \sqsubseteq D$, where $A \in \mathbf{C}$ and there is no axiom $A \equiv C$ in $\mathcal{T}_{\mathrm{def}}$.
- $\mathcal{T}_{\mathrm{g}}$ contains all the remaining axioms from $\mathcal{T}$.

The lazy unfolding optimisation allows the axioms in $\mathcal{T}_{\mathrm{def}}$ and $\mathcal{T}_{\mathrm{sub}}$ to be dealt with more efficiently than those in $\mathcal{T}_{\mathrm{g}}$. Therefore, during the absorption process, FaCT++ processes the axioms in $\mathcal{T}_{\mathrm{g}}$ one at a time, trying to absorb them into $\mathcal{T}_{\mathrm{sub}}$. Those axioms that are not absorbed remain in $\mathcal{T}_{\mathrm{g}}$.

To simplify the formulation of the absorption algorithm, each axiom $C \sqsubseteq D$ is viewed as a clause $\mathbf{G} = \{D, \neg C\}$, corresponding to the axiom $\top \sqsubseteq C \rightarrow D$, which is equivalent to $C \sqsubseteq D$. The concepts in $\mathbf{G}$ are also assumed to be in negation normal form. For each such axiom, FaCT++ applies the absorption steps described in Fig. 3, with $\sqcup(\{C_1, \ldots, C_n\})$ being used to denote $C_1 \sqcup \ldots \sqcup C_n$.

In contrast to the FaCT approach, FaCT++ applies all possible simplifications (except recursive absorption) in a single step. This usually leads to several possible concept and role absorption options, with the intention that heuristics will be used to select the “best” absorption. The development of suitable heuristics is, however, still part of future work.

B Beginning of the absorption cycle.

C Concept absorption. If there is a concept $\neg A \in \mathbf{G}$ such that $A \in \mathbf{C}$ and there is no axiom of the form $A \equiv C$ in $\mathcal{T}_{\text{def}}$, then add $A \sqsubseteq \sqcup(\mathbf{G} \setminus \{\neg A\})$ to $\mathcal{T}_{\text{sub}}$ and exit.

R Role absorption. If there is a concept $\neg\exists R.C \in \mathbf{G}$, then add $\mathsf{Domain}(R, \sqcup(\mathbf{G}))$ to $\mathcal{R}$ and exit.

S Simplification.

1. For every $C \in \mathbf{G}$ such that C is of the form $(C_1 \sqcup \ldots \sqcup C_n)$, change $\mathbf{G}$ as follows: $\mathbf{G} = \mathbf{G} \cup \{C_1, \ldots, C_n\} \setminus \{C\}\}$.
2. For every $A \in \mathbf{G}$ (resp. $\neg A \in \mathbf{G}$), if there is an axiom $A \equiv C$ in $\mathcal{T}_{\text{def}}$, then substitute $A \in \mathbf{G}$ (resp. $\neg A \in \mathbf{G}$) with C (resp. $\sim C$).
3. If any simplification rule was applied, then return to step **B**.

E If there is some $C \in \mathbf{G}$ such that C is of the form $(C_1 \sqcap \ldots \sqcap C_n)$, then for each C_i try to absorb (recursively) $\mathbf{G} \cup \{C_i\} \setminus \{C\}$, and exit. Otherwise, absorption of $\mathbf{G}$ has failed; leave $\mathbf{G}$ in $\mathcal{T}_{\text{g}}$, and exit.

Figure 3: FaCT++ absorption algorithm

Experiments

We have tested FaCT++'s performance when classifying several TBoxes derived from realistic ontologies. In each case range and domain constraints from the ontology had already been transformed into GCIs of the form $\exists R.\top \sqsubseteq C$ and $\top \sqsubseteq \forall R.C$ as described above. All tests used FaCT++ version 0.90 beta running under Linux on an Athlon 2000+ machine with 1Gb of memory.

All our experiments shows that classification time and number of operations reduced by approximately 1 order of magnitude after applying basic role absorption, and by a further 60-80% (approximately) after applying extended role absorption (if avaliable). Due to lack of space we only present here results for a single example.

The RTIMS ontology is taken from a publish and subscribe application where it is used by document publishers to annotate documents so that they can be routed to the appropriate subscribers [13]. The ontology contains about 250 concepts (with medium-complex structure), 76 range and domain constraints and 14 GCIs that are not absorbable by concept absorption.

This ontology is too small to show significant gains in performance. In order to give an indication of the effects of extended role absorption on larger Tboxes containing proportionately more GCIs, we duplicating the RTIMS TBox, systematically renaming concepts and roles, and generated larger TBoxes by unioning together several (from 1 to 100) copies of the the original TBox.

The results of our experiments with these Tboxes are shown in Figure 4, with the problem size (number of copies of the original TBox) on the x-axis and classification time in CPU seconds and number of $\sqcup$-rule applications on the y-axis (using a logarithmic scale). It can be seen that without role absorption the classification time (and other y-axis parameters) increases rapidly with problem size, and without extended (basic) role absorption a TBox con-

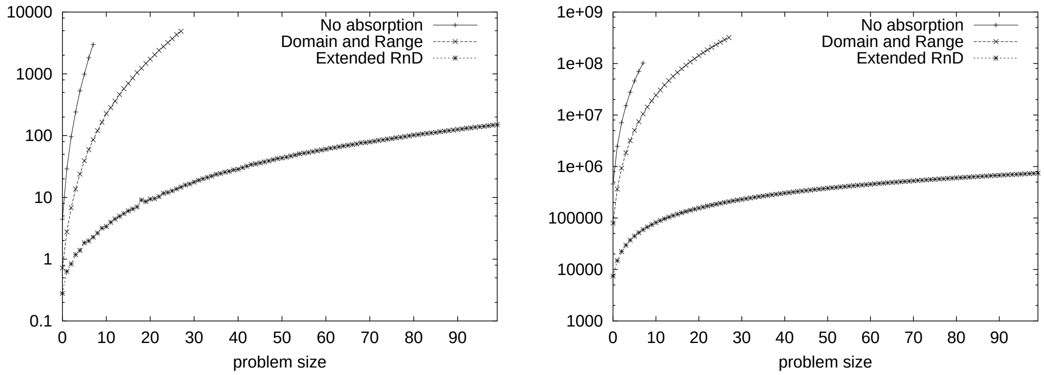

Figure 4: Classification time (left) and ⊔-rule applications (right) for multi-RTIMS TBoxes

sisting of 28 (8) copies of the original already takes several thousand CPU seconds to classify. Further tests failed due to memory limits. In contrast, when using extended role absorption, a TBox consisting of 100 copies of the original could be classified in a little over 100 CPU seconds and requires about 34Mb of memory.

6 Discussion

We have shown how a tableaux algorithm for $\mathcal{SHIQ}$ can be extended to support role boxes that include range and domain axioms, and proved that the extended algorithm is still a decision procedure for the satisfiability and subsumption of $\mathcal{SHIQ}$ concepts w.r.t. such a role box. It should be straightforward to similarly extend tableau algorithms for related DLs such as $\mathcal{SHOQ}$. We have also shown how support for range and domian axioms can be exploited in order to add a new form of absorption optimisation called role absorption.

We have implemented the extended algorithm and the role absorption optimisation in the FaCT++ reasoner, and we have illustrated their effectiveness by analysing the behaviour of FaCT++ when classifying several KBs derived from realistic ontologies. The analysis shows that, not only are the new techniques highly effective, but also that the ordering of different absorption steps can have a significant effect on performance. Future work will include a more detailed study of this effect with a view to devising heuristics that can select the most effective absorption for each GCI.

References

[1] S. Bechhofer, I. Horrocks, C. Goble, and R. Stevens. OilEd: a reason-able ontology editor for the semantic web. In *Proc. of the Joint German/Austrian Conf. on Artificial Intelligence (KI 2001)*, number 2174 in Lecture Notes in Artificial Intelligence, pages 396–408. Springer-Verlag, 2001.

[2] Mike Dean, Dan Connolly, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL web on-

tology language 1.0 reference. W3C Proposed Recommendation, 15 December 2003. Available at `http://www.w3.org/TR/owl-ref/`.

[3] D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, and P. F. Patel-Schneider. OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.

[4] Volker Haarslev and Ralf Möller. High performance reasoning with very large knowledge bases: A practical case study. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)*, pages 161–168, 2001.

[5] Volker Haarslev and Ralf Möller. RACER system description. In *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2001)*, volume 2083 of *Lecture Notes in Artificial Intelligence*, pages 701–705. Springer, 2001.

[6] I. Horrocks. Implementation and optimisation techniques. In Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 306–346. Cambridge University Press, 2003.

[7] Ian Horrocks. Using an expressive description logic: FaCT or fiction? In *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 636–647, 1998.

[8] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. Reviewing the design of DAML+OIL: An ontology language for the semantic web. In *Proc. of the 18th Nat. Conf. on Artificial Intelligence (AAAI 2002)*, pages 792–797. AAAI Press, 2002.

[9] Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Practical reasoning for expressive description logics. In Harald Ganzinger, David McAllester, and Andrei Voronkov, editors, *Proc. of the 6th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR'99)*, number 1705 in Lecture Notes in Artificial Intelligence, pages 161–180. Springer, 1999.

[10] Ian Horrocks and Stephan Tobies. Reasoning with axioms: Theory and practice. In *Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2000)*, pages 285–296, 2000.

[11] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with Protégé-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2001.

[12] Pellet OWL reasoner. Maryland Information and Network Dynamics Lab, `http://www.mindswap.org/2003/pellet/`.

[13] Michael Uschold, Peter Clark, Fred Dickey, Casey Fung, Sonia Smith, Stephen Uczekaj Michael Wilke, Sean Bechhofer, and Ian Horrocks. A semantic infosphere. In Dieter Fensel, Katia Sycara, and John Mylopoulos, editors, *Proc. of the 2003 International Semantic Web Conference (ISWC 2003)*, number 2870 in Lecture Notes in Computer Science, pages 882–896. Springer, 2003.