A Transformation Approach for Classifying $\mathcal{ALCHI}(\mathcal{D})$ Ontologies with a Consequence-based $\mathcal{ALCH}$ Reasoner

Weihong Song, Bruce Spencer, and Weichang Du

Faculty of Computer Science, University of New Brunswick, Fredericton, Canada

Abstract. Consequence-based techniques have been developed to provide efficient classification for less expressive languages. Ontology transformation techniques are often used to approximate axioms in a more expressive language by axioms in a less expressive language. In this paper, we present an approach to use a fast consequence-based $\mathcal{ALCH}$ reasoner to classify an $\mathcal{ALCHI}(\mathcal{D})$ ontology with a subset of OWL 2 datatypes and facets. We transform datatype and inverse role axioms into $\mathcal{ALCH}$ axioms. The transformed ontology preserves sound and complete classification w.r.t the original ontology. The proposed approach has been implemented in the prototype WSClassifier which exhibits the high performance of consequence reasoning. The experiments show that for classifying large and highly cyclic $\mathcal{ALCHI}(\mathcal{D})$ ontologies, WSClassifier's performance is significantly faster than tableau-based reasoners.

1 Introduction

Ontology classification is the foundation of many ontology reasoning tasks. Recently, consequence-based techniques have been developed to provide efficient classification for sublanguages of OWL 2 DL profile, e.g. $\mathcal{EL}^{++}$ [2,3,8], Horn-$\mathcal{SHIQ}$ [7], $\mathcal{EL}^{\perp}(\mathcal{D})$ [9], $\mathcal{ALCH}$ [13]. There have been some approaches to use existing consequence-based reasoners to classify more expressive ontologies, like MORe [1]. In this paper, we propose an approach to use a consequence-based $\mathcal{ALCH}$ reasoner to classify an $\mathcal{ALCHI}(\mathcal{D})$ ontology by transforming it into an $\mathcal{ALCH}$ ontology with soundness and completeness preserved. The purpose of the approach is to extend the expressiveness of the existing consequence-based reasoner without changing its complex inference rules and implementation. All proofs and further technical details can be found in our technical report [15].

Ontology transformation is often accomplished by approximating non-Horn ontologies/theories by Horn replacements [12,11,16]. These approximations can be used to optimize reasoning by exploiting more efficient inference for Horn ontologies/theories. The approximation $\mathcal{O}'$ in Ren *et al.* [11] is a lower bound of the original ontology $\mathcal{O}$, i.e. $\mathcal{O}'$ entails no more subsumptions than $\mathcal{O}$ does. In contrast, approximation in Zhou *et al.* [16] provides an upper bound. Kautz *et al.* [12] computes both upper and lower bounds of propositional logic theories. Another approach preserves both soundness and completeness of classification results such as the elimination of transitive roles in Kazakov [7]. Our work of this paper is of the second kind. We classify an $\mathcal{ALCHI}(\mathcal{D})$ ontology $\mathcal{O}$ in two stages: (1) transform $\mathcal{O}$ into an $\mathcal{ALCHI}$ ontology $\mathcal{O}^-_{\mathcal{D}}$ s.t. $\mathcal{O} \models A \sqsubseteq B$ iff $\mathcal{O}^-_{\mathcal{D}} \models A \sqsubseteq B$;

(2) transform $\mathcal{O}_{\mathcal{D}}^{-}$ into an $\mathcal{ALCH}$ ontology $\mathcal{O}_{\mathcal{ID}}^{-}$ s.t. $\mathcal{O}_{\mathcal{ID}}^{-} \models A \sqsubseteq B$ iff $\mathcal{O}_{\mathcal{D}}^{-} \models A \sqsubseteq B$. We use these approaches to implement a reasoner called WSClassifier which transforms an $\mathcal{ALCHI}(\mathcal{D})$ ontology into an $\mathcal{ALCH}$ ontology and classifies it with a fast consequence $\mathcal{ALCH}$ reasoner ConDOR [13]. WSClassifier is significantly faster than tableau-based reasoners on large and highly cyclic ontologies.

In our previous work [14] we approximated an $\mathcal{ALCHOI}$ ontology by an $\mathcal{ALCH}$ ontology which was then classified by a hybrid of consequence- and tableau-based reasoners. Unlike [14], in this paper we claim completeness for $\mathcal{I}$'s transformation. Calvanese *et al.* [4] introduces a general approach to eliminate inverse roles and functional restrictions from $\mathcal{ALCFI}$ to $\mathcal{ALC}$. For eliminating $\mathcal{I}$, the approach needs to add one axiom for each inverse role and each concept. So the number of axioms added can be very large. Ding *et al.* [6] introduces a new mapping from $\mathcal{ALCI}$ to $\mathcal{ALC}$ and further extends it to a mapping from $\mathcal{SHI}$ to $\mathcal{SH}$ in [5]. The approach allows tableau-based decision procedures to use some caching techniques and improve the reasoning performance in practice. Both approaches in [4,5] preserve the soundness and completeness of inference after elimination of $\mathcal{I}$. Our approach is similar to the one in [6,5]. However, the NNF normalized form in [6,5] in which $\top$ appears in the left side of all axioms will dramatically degrade the performance of our consequence-based $\mathcal{ALCH}$ reasoner. Thus we eliminate the inverse role based on our own normalized form and our approach is more suitable for consequence-based reasoners.

2 Preliminary

Due to space limitation, we only list the most necessary syntax and semantics of $\mathcal{ALCHI}(\mathcal{D})$ in the paper, the complete illustration can be found in our Technical Report [15]. The syntax of $\mathcal{ALCHI}(\mathcal{D})$ uses atomic concepts N_C, atomic roles N_R and features N_F. We use A, B for atomic concepts, C, D for concepts, r, s for atomic roles, R, S for roles, F, G for features. The parameter $\mathcal{D}$ defines a *datatype map* $\mathcal{D} = (N_{DT}, N_{LS}, N_{FS}, \cdot^{\mathcal{D}})$, where: (1) N_{DT} is a set of datatype names; (2) N_{LS} is a function assigning to each $d \in N_{DT}$ a set of constants $N_{LS}(d)$; (3) N_{FS} is a function assigning to each $d \in N_{DT}$ a set of facets $N_{FS}(d)$, each $f \in N_{FS}(d)$ has the form (p_f, v); (4) $\cdot^{\mathcal{D}}$ is a function assigning a datatype interpretation $d^{\mathcal{D}}$ to each $d \in N_{DT}$ called the *value space* of d, a data value $v^{\mathcal{D}} \in d^{\mathcal{D}}$ for each $v \in N_{LS}(d)$, and a facet interpretation $f^{\mathcal{D}}$ for each $f \in \bigcup_{d \in N_{DT}} N_{FS}(d)$. Since one facet may be shared by multiple datatypes, we define its interpretation as containing subsets of all the relevant datatypes. $\top_{\mathcal{D}}$, d, $d[f]$ or $\{v\}$ are basic forms of data ranges, which we call *atomic data range*s. A data range dr is defined recursively using $\sqcap$, $\sqcup$, and $\neg$. A role R is either an atomic role r or *inverse role* r^{-}. Semantics of $\mathcal{ALCHI}(\mathcal{D})$ is defined via an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \Delta^{\mathcal{D}}, \cdot^{\mathcal{I}}, \cdot^{\mathcal{D}})$. $\Delta^{\mathcal{I}}$ and $\Delta^{\mathcal{D}}$ are disjoint non-empty sets called *object domain* and *data domain*. $d^{\mathcal{D}} \subseteq \Delta^{\mathcal{D}}$ for each $d \in N_{DT}$. $\cdot^{\mathcal{I}}$ assigns a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to each $A \in N_C$, a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to each $r \in N_R$ and a relation $F^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{D}}$ to each $F \in N_F$. $F^{\mathcal{I}}(x) = \{v \mid (x, v) \in F^{\mathcal{I}}\}$. $\cdot^{\mathcal{D}}$ interprets data ranges and concepts, as shown in Table 1. We write $N_{DT}(\mathcal{O})$ and $ADR(\mathcal{O})$ for all datatypes and atomic data ranges in $\mathcal{O}$. And $ADR_d(\mathcal{O})$ denotes the subset of $ADR(\mathcal{O})$ in datatype d, i.e. of the form d, $d[f]$ or $\{v\}$ where $v \in N_{LS}(d)$.

Table 1. Part of Model Theoretic Semantics of $\mathcal{ALCHI}(\mathcal{D})$

Semantics of Data Ranges		
$(\top_{\mathcal{D}})^{\mathcal{D}} = \Delta^{\mathcal{D}}$	$\{v\}^{\mathcal{D}} = \{v^{\mathcal{D}}\}$	$(dr_1 \sqcap dr_2)^{\mathcal{D}} = dr_1^{\mathcal{D}} \cap dr_2^{\mathcal{D}}$
$(d[f])^{\mathcal{D}} = d^{\mathcal{D}} \cap f^{\mathcal{D}}$	$(\neg dr)^{\mathcal{D}} = \Delta^{\mathcal{D}} \setminus dr^{\mathcal{D}}$	$(dr_1 \sqcup dr_2)^{\mathcal{D}} = dr_1^{\mathcal{D}} \cup dr_2^{\mathcal{D}}$
Semantics of Concepts, Roles and Axioms		
$F \sqsubseteq G \Rightarrow F^{\mathcal{I}} \subseteq G^{\mathcal{I}}$	$(\exists F.dr)^{\mathcal{I}} = \{x \mid F^{\mathcal{I}}(x) \cap dr^{\mathcal{D}} \neq \emptyset\}$	$(\forall F.dr)^{\mathcal{I}} = \{x \mid F^{\mathcal{I}}(x) \subseteq dr^{\mathcal{D}}\}$

3 Transformation for Datatypes

In this section we introduce how we transform an $\mathcal{ALCHI}(\mathcal{D})$ ontology $\mathcal{O}$ into an $\mathcal{ALCHI}$ ontology $\mathcal{O}_{\mathcal{D}}^-$ such that $\mathcal{O} \models A \sqsubseteq B$ iff $\mathcal{O}_{\mathcal{D}}^- \models A \sqsubseteq B$. We assume all the datatypes in $\mathcal{D}$ are disjoint, as do Motik et al [10]. We apply our approach to some commonly used datatypes: (1) real with facets rational, decimal, integer, $>_a$, $\geq_a$, $<_a$ and $\leq_a$; (2) strings with equal value; (3) boolean values.

Our basic idea to produce $\mathcal{O}_{\mathcal{D}}^-$ from $\mathcal{O}$ is to encode features into roles and data ranges into concepts, and then add extra axioms to preserve the subsumptions between atomic concepts in $N_C(\mathcal{O})$. Table 2 gives the definition of encoding function φ over atomic elements in $\mathcal{O}$, where A_d, A_f, A_v are fresh concepts and R_F is a fresh role. φ over complex data ranges, roles, concepts and axioms are defined recursively using corresponding constructors. It is easy to prove that classification of $\varphi(\mathcal{O}) = \{\varphi(\alpha) \mid \alpha \in \mathcal{O}\}$ is sound w.r.t. $\mathcal{O}$ (proof see [15]). In order to preserve classification completeness, extra axioms need to be added to $\varphi(\mathcal{O})$ to get $\mathcal{O}_{\mathcal{D}}^-$. Algorithm 1 shows how $\mathcal{O}_{\mathcal{D}}^-$ is computed. In the procedure

Table 2. Encoding φ for atomic concepts/roles/features/data ranges

$\varphi(\top_{\mathcal{D}}) = \top$	$\varphi(d[f]) = A_d \sqcap A_f$	$\varphi(\top) = \top$	$\varphi(A) = A$
$\varphi(d) = A_d$	$\varphi(\{v\}) = A_v$	$\varphi(R) = R$	$\varphi(F) = R_F$

we use two functions $\mathsf{normalize}_d$ and $\mathsf{getAxioms}_d$ for each datatype $d \in N_{DT}(\mathcal{O})$. $\mathsf{normalize}_d$ rewrites data ranges $d[f]$ into normalized forms to reduce the kinds of facets used. $\mathsf{getAxioms}_d$ produces a set of $\mathcal{ALCHI}$ axioms $\mathcal{O}_d^+$ to be included into $\mathcal{O}_{\mathcal{D}}^-$. Details will be explained later for the datatypes and facets supported. In order to preserve classification completeness w.r.t. $\mathcal{O}$, $\mathsf{getAxioms}_d$ must generate axioms explicitly showing the relationships implicit among data ranges before encoding, i.e., the **data-range-relationship-preserving property**: for any $ar_1, \ldots, ar_n, ar'_1, \ldots, ar'_m \in ADR_d(\mathcal{O})$, if $\left((\bigsqcap_{i=1}^n ar_i) \sqcap (\bigsqcap_{j=1}^m \neg ar'_j)\right)^{\mathcal{D}} = \emptyset$, then $(\bigsqcap_{i=1}^n \varphi(ar_i)) \sqcap (\bigsqcap_{j=1}^m \neg\varphi(ar'_j))$ is unsatisfiable in $\mathcal{O}_d^+ = \mathsf{getAxioms}_d(ADR_d(\mathcal{O}), \varphi)$. We prove this condition is sufficient for completeness in [15].

For boolean type, we do not have any facets, so $\mathsf{normalize}_d$ does nothing. Since the only atomic data ranges are $xsd\!:\!boolean$, $\{true\}$ and $\{false\}$, $\mathsf{getAxioms}_d$ only needs to return two axioms $\varphi(xsd : boolean) \equiv \varphi(\{true\}) \sqcup \varphi(\{false\})$ and $\varphi(\{true\}) \sqcap \varphi(\{false\}) \sqsubseteq \bot$. For string type, currently we do not support any facets, so $\mathsf{normalize}_d$ does nothing either. Atomic data ranges are ei-

Algorithm 1: Datatype Transformation

Input: An $\mathcal{ALCHI}(\mathcal{D})$ ontology $\mathcal{O}$
Output: An $\mathcal{ALCHI}$ ontology $\mathcal{O}_{\mathcal{D}}^{-}$ with the same classification result as $\mathcal{O}$

foreach $d \in N_{DT}(\mathcal{O})$ **do**
  foreach $adr \in ADR_d(\mathcal{O})$ **do**
    Replace adr with $\mathsf{normalize}_d(adr)$ in $\mathcal{O}$;
Create an encoding φ for $\mathcal{O}$ and initialize $\mathcal{O}_{\mathcal{D}}^{-}$ with $\varphi(\mathcal{O})$;
foreach $d_1, d_2 \in N_{DT}(\mathcal{O}), d_1 \neq d_2$ **do** $\mathcal{O}_{\mathcal{D}}^{-} \leftarrow \mathcal{O}_{\mathcal{D}}^{-} \cup \{\varphi(d_1) \sqcap \varphi(d_2) \sqsubseteq \bot\}$;
foreach $d \in N_{DT}(\mathcal{O})$ **do** $\mathcal{O}_{\mathcal{D}}^{-} \leftarrow \mathcal{O}_{\mathcal{D}}^{-} \cup \mathsf{getAxioms}_d(ADR_d(\mathcal{O}), \varphi)$;
return $\mathcal{O}_{\mathcal{D}}^{-}$;

ther $xsd:string$ or of the form $\{c\}$, where c is a constant. We need to add $\varphi(\{c\}) \sqsubseteq \varphi(xsd:string)$ for each $\{c\} \in ADR_{\mathbb{R}}(\mathcal{O})$, as well as pairwise disjoint axioms for all such $\varphi(\{c\})$. Numeric datatypes are the most commonly used datatypes in ontologies. Here we discuss the implementation for $owl:real$, which we denote by $\mathbb{R}$. $owl:rational$, $xsd:decimal$ and $xsd:integer$ are treated as facets rat, dec and int of $\mathbb{R}$, respectively. Comparison facets of the forms $>_a$, $<_a$, $\geq_a$, $\leq_a$ are supported. For $\mathsf{normalize}_d$ with input ar, we need: (1) if $adr = \mathbb{R}[f]$, transform it to equivalent data ranges using only facets of the form $>_a$, e.g. $\mathbb{R}[<_a] = \mathbb{R} \sqcap \neg(\mathbb{R}[>_a] \sqcup \{a\})$; (2) replace any constant a used in ar with a normal form, so that any constants having the same interpretation becomes the same after normalization, e.g. integer constants +3 and 3 are both interpreted as real number 3, so they are normalized into the same form "3"^^$xsd:integer$. Algorithm 2 gives the details of $\mathsf{getAxioms}_{\mathbb{R}}$ for real numbers. For boolean and string, it is obvious that the corresponding $\mathsf{getAxioms}_d$ has data-range-relationship-preserving property. For $\mathsf{getAxioms}_{\mathbb{R}}$, we prove this property in [15]. So if $\mathcal{O} \models A \sqsubseteq B$, then $\mathcal{O}_{\mathcal{D}}^{-} \models A \sqsubseteq B$.

4 Transformation for Inverse Roles

In this section, we discuss how we transform an $\mathcal{ALCHI}$ ontology $\mathcal{O}_{\mathcal{D}}^{-}$ into an $\mathcal{ALCH}$ ontology $\mathcal{O}_{\mathcal{ID}}^{-}$, such that $\mathcal{O}_{\mathcal{D}}^{-} \models A \sqsubseteq B$ iff $\mathcal{O}_{\mathcal{ID}}^{-} \models A \sqsubseteq B$(proof see [15]). Algorithm 3 shows the details of transformation for inverse roles. In the procedure Inv_r contains the set of atomic roles which are inverses of r. Line 1 initializes $\mathcal{O}_{\mathcal{ID}}^{-}$ with $\mathcal{ALCH}$ axioms in $\mathcal{O}_{\mathcal{D}}^{-}$. Lines 2 to 6 initializes Inv_r and put all r where $\mathsf{Inv}_r \neq \emptyset$ into $\mathsf{RolesToBeProcessed}$. Lines 7 to 16 processes each role in $\mathsf{RolesToBeProcessed}$ and adds axioms into $\mathcal{O}_{\mathcal{ID}}^{-}$ to address the effect of inverse role axioms. Detail explanations of Algorithm 3 are in our technical report [15].

5 Experiment and Conclusion

In experiments we compare the runtime of our WSClassifier with all other available $\mathcal{ALCHI}(\mathcal{D})$ reasoners HermiT, Fact++ and Pellet, which all happen to be tableau-based reasoners. We use all large and highly cyclic ontologies we can

Algorithm 2: getAxioms$_{\mathbb{R}}$ for $\mathbb{R}$

Input: A set of atomic data ranges $ADR_{\mathbb{R}}(\mathcal{O})$ of type $\mathbb{R}$, encoding function φ
Output: A set of axioms $\mathcal{O}^+_{\mathbb{R}}$

$\mathcal{O}^+_{\mathbb{R}} \leftarrow \emptyset$;
foreach $\{v\} \in ADR_{\mathbb{R}}(\mathcal{O})$ **do**
  if $\mathbb{R}[int] \in ADR_{\mathbb{R}}(\mathcal{O})$ *and* $v^{\mathcal{D}} \in (\mathbb{R}[int])^{\mathcal{D}}$ **then** add $\varphi(\{v\}) \sqsubseteq \varphi(int)$ to $\mathcal{O}^+_{\mathbb{R}}$;
  if $\mathbb{R}[dec] \in ADR_{\mathbb{R}}(\mathcal{O})$ *and* $v^{\mathcal{D}} \in (\mathbb{R}[dec])^{\mathcal{D}}$ **then** add $\varphi(\{v\}) \sqsubseteq \varphi(dec)$ to $\mathcal{O}^+_{\mathbb{R}}$;
  if $\mathbb{R}[rat] \in ADR_{\mathbb{R}}(\mathcal{O})$ *and* $v^{\mathcal{D}} \in (\mathbb{R}[rat])^{\mathcal{D}}$ **then** add $\varphi(\{v\}) \sqsubseteq \varphi(rat)$ to $\mathcal{O}^+_{\mathbb{R}}$;
if $\mathbb{R}[int], \mathbb{R}[dec] \in ADR_{\mathbb{R}}(\mathcal{O})$ **then** add $\varphi(int) \sqsubseteq \varphi(dec)$ to $\mathcal{O}^+_{\mathbb{R}}$;
if $\mathbb{R}[int], \mathbb{R}[rat] \in ADR_{\mathbb{R}}(\mathcal{O})$ **then** add $\varphi(int) \sqsubseteq \varphi(rat)$ to $\mathcal{O}^+_{\mathbb{R}}$;
if $\mathbb{R}[dec], \mathbb{R}[rat] \in ADR_{\mathbb{R}}(\mathcal{O})$ **then** add $\varphi(dec) \sqsubseteq \varphi(rat)$ to $\mathcal{O}^+_{\mathbb{R}}$;
Put all $\mathbb{R}[>_a] \in ADR_{\mathbb{R}}(\mathcal{O})$ in *fArray* with ascending order of a;
foreach *pair of adjacent elements* $\mathbb{R}[>_a]$ *and* $\mathbb{R}[>_b]$ *($a < b$) in fArray* **do**
  add $\varphi(>_b) \sqsubseteq \varphi(>_a)$ to $\mathcal{O}^+_{\mathbb{R}}$;
  if $\mathbb{R}[int] \in ADR_{\mathbb{R}}(\mathcal{O})$ **then**
    $M \leftarrow \{\{a_i\}\}_{i=1}^n$, where $a_1, \ldots, a_n$ are all integer constants in $(a, b]$;
    if $M \subseteq ADR_{\mathbb{R}}(\mathcal{O})$ **then**
      add $\varphi(int) \sqcap \varphi(>_a) \sqcap \neg\varphi(>_b) \sqcap (\bigsqcap_{i=1}^n \neg\varphi(\{a_i\})) \sqsubseteq \bot$ to $\mathcal{O}^+_{\mathbb{R}}$;
  Let N be all v such that $\{v\} \in ADR_{\mathbb{R}}(\mathcal{O})$ and $v^{\mathcal{D}} \in (a, b]$;
  foreach $v \in N$ **do** add $\varphi(\{v\}) \sqsubseteq \varphi(>_a)$, $\varphi(\{v\}) \sqcap \varphi(>_b) \sqsubseteq \bot$ to $\mathcal{O}^+_{\mathbb{R}}$;
  foreach $v_1, v_2 \in N, v_1 \neq v_2$ **do** add $\varphi(\{v_1\}) \sqcap \varphi(\{v_2\}) \sqsubseteq \bot$ to $\mathcal{O}^+_{\mathbb{R}}$;
return $\mathcal{O}^+_{\mathbb{R}}$;

access to. FMA-constitutionalPartForNS(FMA-C) is the only large and highly cyclic ontology that contains $\mathcal{ALCHI}(\mathcal{D})$ constructors. We remove seven axioms using $xsd:float$. For Full-Galen which language is $\mathcal{ALEHIF}+$ without "D", we introduce some new data type axioms by converting some axioms using roles hasNumber and hasMagnitude into axioms with new features hasNumberDT and hasMagnitudeDT. Some concepts which should be modeled as data ranges are also converted to data ranges. Wine is a small but cyclic ontology. We also include two commonly used ontologies ACGT and OBI which are not highly cyclic. For Wine, ACGT and OBI, we change xsd:int, xsd:positiveInteger, xsd:nonNegativeInteger to xsd:integer, xsd:float to owl:rational, and remove xsd:dateTime if applicable. For all the ontologies, we reduce their language to $\mathcal{ALCHI}(\mathcal{D})$. The ontologies are available from our website.[1].The experiments were conducted on a laptop with Intel Core i7-2670QM 2.20GHz quad core CPU and 16GB RAM. We set the Java heap space to 12GB and the time limit to 24 hours.

Table 3 summarizes the result. HermiT is set to configuration with simple core blocking and individual reuse. WSClassifier is significantly faster than the tableau-based reasoners on the three highly cyclic large ontologies Galen-Heart, Full-Galen and FMA-C. ACGT is not highly cyclic, but WSClassifier is still faster. For the other two ontologies where WSClassifier is not the fastest, Wine is cyclic but small, OBI is not highly cyclic. The classification time for them on

[1] `http://isel.cs.unb.ca/~wsong/ORE2013WSClassifierOntologies.zip`

Algorithm 3: Transformation for inverse roles

Input: Normalized ontology $\mathcal{ALCHI}$ ontology $\mathcal{O}_{\mathcal{D}}^{-}$
Output: An $\mathcal{ALCH}$ ontology $\mathcal{O}_{\mathcal{ID}}^{-}$ having the same classification result as $\mathcal{O}_{\mathcal{D}}^{-}$
Initialize $\mathcal{O}_{\mathcal{ID}}^{-}$ with all $\mathcal{ALCH}$ axioms in $\mathcal{O}_{\mathcal{D}}^{-}$, excluding inverse role axioms;
foreach $r \in N_R(\mathcal{O})$ **do** $\mathsf{Inv}_r \leftarrow \emptyset$;
$\mathsf{RolesToBeProcessed} \leftarrow \emptyset$;
foreach $r' = r^- \in \mathcal{O}_{\mathcal{D}}^{-}$ **do**
  $\mathsf{Inv}_r \leftarrow \mathsf{Inv}_r \cup \{r'\}$; $\mathsf{Inv}_{r'} \leftarrow \mathsf{Inv}_{r'} \cup \{r\}$;
  $\mathsf{RolesToBeProcessed} \leftarrow \mathsf{RolesToBeProcessed} \cup \{r, r'\}$;
while $\mathsf{RolesToBeProcessed} \neq \emptyset$ **do**
  remove a role r from $\mathsf{RolesToBeProcessed}$ and pick a role r' from Inv_r;
  foreach $r^* \in \mathsf{Inv}_r$ *where* r^* *is not* r' **do** add $r' \equiv r^*$ to $\mathcal{O}_{\mathcal{ID}}^{-}$;
  foreach $r \sqsubseteq s \in \mathcal{O}_{\mathcal{D}}^{-}$ **do**
    if $\mathsf{Inv}_s = \emptyset$ **then**
      add a fresh atomic role s' to Inv_s;
      $\mathsf{RolesToBeProcessed} \leftarrow \mathsf{RolesToBeProcessed} \cup \{s\}$;
    pick a role s' from Inv_s and add $r' \sqsubseteq s'$ to $\mathcal{O}_{\mathcal{ID}}^{-}$;
  foreach $\exists r.A \sqsubseteq B \in \mathcal{O}_{\mathcal{D}}^{-}$ **do** add $A \sqsubseteq \forall r'.B$ to $\mathcal{O}_{\mathcal{ID}}^{-}$;
  foreach $A \sqsubseteq \forall r.B \in \mathcal{O}_{\mathcal{D}}^{-}$ **do** add $\exists r'.A \sqsubseteq B$ to $\mathcal{O}_{\mathcal{ID}}^{-}$;
return $\mathcal{O}_{\mathcal{ID}}^{-}$

all reasoners are significantly shorter comparing with the time on large highly cyclic ontologies. Then WSClassifier took a larger percentage of time on the overhead to transmit the ontology to and from ConDOR.

Table 3. Comparison of classification performance of $\mathcal{ALCHI}(\mathcal{D})$ ontologies

	HermiT	**Pellet**	**FaCT++**	**WSClassifier**
Wine	1.160 sec	0.430 sec	0.005 sec	0.400 sec
ACGT	9.603 sec	2.955 sec	*	1.945 sec
OBI	3.166 sec	45.261 sec	*	8.835 sec
Galen-Heart	123.628 sec	–	–	2.779 sec
Full-Galen	–	–	–	16.774 sec
FMA-C	–	–	–	32.74 sec

Note: "–": out of time or memory "*": some datatypes are not supported

We have transformed some commonly used OWL 2 datatypes and facets and inverse role axioms in an $\mathcal{ALCHI}(\mathcal{D})$ ontology to $\mathcal{ALCH}$ and classified it on an $\mathcal{ALCH}$ reasoner with soundness and completeness of classification preserved. WSClassifier greatly outperforms tableau-based reasoners when the ontologies are large and highly cyclic. Future work includes extension to other data types and facets, and further optimization, e.g. adapting the idea of Magka *et al.* [9] to WSClassifier to distinguish positive and negative occurrences of data ranges, in order to reduce the number of axioms to be added.

References

1. Armas Romero, A., Cuenca Grau, B., Horrocks, I.: MORe: Modular combination of OWL reasoners for ontology classification. In: Proc. of ISWC. pp. 1–16 (2012)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: IJCAI. pp. 364–369 (2005)
3. Baader, F., Lutz, C.: Pushing the $\mathcal{EL}$ envelope further. In: Proc. of OWLED (2008)
4. Calvanese, D., De Giacomo, G., Rosati, R.: A note on encoding inverse roles and functional restrictions in alc knowledge bases. In: Proc. of the 5th Int. Description Logic Workshop. DL. vol. 98, pp. 11–20 (1998)
5. Ding, Y.: Tableau-based reasoning for description logics with inverse roles and number restrictions. http://users.encs.concordia.ca/~haarslev/students/Yu_Ding.pdf (2008)
6. Ding, Y., Haarslev, V., Wu, J.: A new mapping from alci to alc. In: Proc. DL-2007. CEUR Workshop Proceedings. vol. 250. Citeseer (2007)
7. Kazakov, Y.: Consequence-driven reasoning for Horn $\mathcal{SHIQ}$ ontologies. In: Proc. of IJCAI. pp. 2040–2045 (2009)
8. Kazakov, Y., Krötzsch, M., Simančík, F.: Concurrent classification of $\mathcal{EL}$ ontologies. In: Proc. of ISWC. pp. 305–320 (2011)
9. Magka, D., Kazakov, Y., Horrocks, I.: Tractable extensions of the description logic $\mathcal{EL}$ with numerical datatypes. J. Automated Reasoning 47(4), 427–450 (2011)
10. Motik, B., Horrocks, I.: Owl datatypes: Design and implementation. In: International Semantic Web Conference. pp. 307–322 (2008)
11. Ren, Y., Pan, J.Z., Zhao, Y.: Soundness preserving approximation for TBox reasoning. In: Proc. of AAAI (2010)
12. Selman, B., Kautz, H.: Knowledge compilation and theory approximation. J. ACM 43(2), 193–224 (1996)
13. Simančík, F., Kazakov, Y., Horrocks, I.: Consequence-based reasoning beyond Horn ontologies. In: Proc. of IJCAI. pp. 1093–1098 (2011)
14. Song, W., Spencer, B., Du, W.: WSReasoner: A prototype hybrid reasoner for $\mathcal{ALCHOI}$ ontology classification using a weakening and strengthening approach. In: Proc. of the 1st Int. OWL Reasoner Evaluation Workshop (2012)
15. Song, W., Spencer, B., Du, W.: Technical report of a transformation approach for classifying $\mathcal{ALCHI}(\mathcal{D})$ ontologies with a consequence-based $\mathcal{ALCH}$ reasoner. Tech. rep. (2013), http://www.cs.unb.ca/tech-reports/documents/TR13-225.pdf
16. Zhou, Y., Cuenca Grau, B., Horrocks, I.: Efficient upper bound computation of query answers in expressive description logics. In: Proc. of DL (2012)