Front. Comput. Sci., 2019, 13(1): 139–156
https://doi.org/10.1007/s11704-016-6397-2

RESEARCH ARTICLE

ComR: a combined OWL reasoner for ontology classification

Changlong WANG[1,2,3], Zhiyong FENG[1,2], Xiaowang ZHANG (✉)[1,2], Xin WANG[1,2], Guozheng RAO[1,2], Daoxun FU[1,2]

1 School of Computer Science and Technology, Tianjin University, Tianjin 300072, China
2 Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin 300072, China
3 School of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China

Abstract Ontology classification, the problem of computing the subsumption hierarchies for classes (atomic concepts), is a core reasoning service provided by Web Ontology Language (OWL) reasoners. Although general-purpose OWL 2 reasoners employ sophisticated optimizations for classification, they are still not efficient owing to the high complexity of tableau algorithms for expressive ontologies. Profile-specific OWL 2 EL reasoners are efficient; however, they become incomplete even if the ontology contains only a small number of axioms that are outside the OWL 2 EL fragment. In this paper, we present a technique that combines an OWL 2 EL reasoner with an OWL 2 reasoner for ontology classification of expressive *SROIQ*. To optimize the workload, we propose a task decomposition strategy for identifying the minimal non-EL subontology that contains only necessary axioms to ensure completeness. During the ontology classification, the bulk of the workload is delegated to an efficient OWL 2 EL reasoner and only the minimal non-EL subontology is handled by a less efficient OWL 2 reasoner. The proposed approach is implemented in a prototype *ComR* and experimental results show that our approach offers a substantial speedup in ontology classification. For the well-known ontology NCI, the classification time is reduced by 96.9% (resp. 83.7%) compared against the standard reasoner Pellet (resp. the modular reasoner MORe).

Keywords OWL, ontology, classification, reasoner

Received August 2, 2016; accepted December 7, 2016

E-mail: xiaowangzhang@tju.edu.cn

1 Introduction

Ontologies expressed in the Web Ontology Language (OWL) [1, 2] and its second version (OWL 2) [3, 4] play a central role in the development of the Semantic Web [5]. They are also widely used in biomedical information systems [6–8] and other areas, such as agriculture [9], astronomy [10], defense [11], geography [12], and traffic [13, 14].

Ontology-based information systems require efficient and robust reasoning services. One of the main reasoning tasks is ontology classification, which requires computing all entailed (implicit) subsumption relations between the classes (atomic concepts) in an ontology and thus allows one to construct the terminology in the form of a subsumption hierarchy [15]. This hierarchy provides useful information on the connection between different concepts and it can also be used to speed up other inference services. For example, classification facilitates instance checking. If the individual i is an instance of the concept C, to decide whether i is an instance of a concept D, it should be sufficient to check whether C is subsumed by D. Consequently, off-line classification of the ontology may provide a way to precompute instance checks, providing an on-line speedup. The classification results are also useful for navigating an ontology and identifying modeling errors, as well as for explaining and query answering [16].

The underpinning logic of OWL is description logic (DL) [17]: OWL DL is based on the DL *SHOIN*, and its revision OWL 2 is based on the more expressive DL *SROIQ*. It is well known that expressive DLs have high worst-case com-

putational complexity. For example, classification in the DL *SROIQ* is N2EXPTIME-complete [18], i.e., it is decided by a nondeterministic Turing machine in $O(2^{2^{p(n)}})$ time, where n is the size of a given ontology, and $p(n)$ is a polynomial function of n. To classify ontologies in applications, the current OWL 2 reasoners, which are based on tableau [19] or hypertableau [20] algorithms and build models to check concept satisfiability, have provided some dedicated optimizations for ontology classification. For example, HermiT [21] uses hypertableau calculus that completely avoids subsumption tests for deterministic ontologies; subsumption can be just read out of the models produced for concept satisfiability tests. FaCT++ [22] reduces the number of subsumption tests for completely defined concepts, and an extension of this optimization with structural pseudo-model embedding has been used by RacerPro [23]. The latest version of Pellet [24] can apparently switch to a specialized procedure when the ontology is within a fragment of OWL 2 EL. As each reasoner employs a different set of algorithms and optimization techniques, different reasoners may exhibit significantly different performance on the same ontology [25].

In spite of using many sophisticated optimizations, general-purpose OWL 2 reasoners are still not able to efficiently classify very large ontologies in the NCBO BioPortal ontology repository, such as Protein and NCI. To obtain favorable computational properties, researchers have paid more attention to other tractable ontology languages, i.e., OWL 2 EL, OWL 2 RL, and OWL 2 QL [26], which are called the profiles of OWL 2. In the three profiles, the OWL 2 EL profile is based on the lightweight DL EL^{++}, for which most standard reasoning tasks can be performed in polynomial time [27]. Moreover, the OWL 2 EL is very suitable for representing terminologies in the area of biomedicine, and most of the ontologies in the NCBO BioPortal are expressed in this language, such as gene ontology (GO) [28] and SNOMED CT [29], and the well-known ontology—the Foundational Model of Anatomy (FMA) [10]. Several very efficient profile-specific reasoners have been developed for the OWL 2 EL, such as CEL [30], jcel [31], and ELK [32]. These reasoners use many so-called completion rules (a variant of saturation) [27] to derive new concept inclusions, and they are often able to outperform tableau-based reasoners. In particular, profile-specific reasoners allow a fast one-pass handling of several reasoning tasks such as classification, whereas tableau-based reasoners perform classification by a pairwise comparison of the atomic concepts. Unfortunately, the OWL 2 EL reasoners are restricted to the OWL 2 EL fragment, and if an ontology contains axioms beyond this fragment, then the OWL 2 EL reasoners may become incomplete for reasoning.

However, in practical ontologies, some concepts or roles cannot be precisely expressed in only OWL 2 EL. Hence, ontology engineers must extend ontologies with some axioms that use features of more expressive DLs, e.g., disjunction ("$\sqcup$"). Consequently, these ontologies often contain a few axioms that are outside the lightweight DL EL^{++}. For example, many commonly used biomedical ontologies, such as Open Biomedical Ontologies (OBO) [33], are covered by the OWL 2 EL to a large degree. Of the 226,038 axioms in the version *V14.03e* of the NCI ontology [34], only 67 are outside the OWL 2 EL fragment, and of the 46,698 axioms in the Protein ontology [6], only 16 are outside the OWL 2 EL fragment. For these ontologies, the efficient OWL 2 EL reasoning system cannot reason completely. The tableau-based OWL 2 reasoners, on the other hand, are not limited to a specific OWL profile, but even highly optimized OWL 2 reasoners might not be efficient enough to handle large ontologies, such as NCI. Hence, ontology engineers must decide whether they only use the restricted features of certain language fragments such that their ontologies can be handled by specialized reasoners or they must face possible performance losses by using reasoning systems that are more general.

To cope with such ontologies that contain a small number of axioms beyond the OWL 2 profiles, new approaches are proposed to improve the reasoning performance by (i) combining efficient profile-specific reasoners with fully-fledged tableau reasoners in a black box manner [35–37], or (ii) adapting the tableau algorithms and completion-based procedures, and then coupling them in a new reasoner [38]. Besides, the Stanford Natural Language Processing Group has proposed an approach for incorporating both of these signals in a unified framework for question answering analogously [39]. The common idea of those approaches is to delegate as much work as possible to the specialized and efficient procedure, which allows reducing the workload of the less efficient tableau-algorithm. However, those approaches have some drawbacks: the black-box approach, such as [35], delegates considerable workload to the less efficient OWL 2 reasoner; the tight coupling approach [38] needs to modify the existing reasoners.

In this paper, we propose an approach that combines a special OWL 2 EL reasoner with a general-purpose OWL 2 reasoner to classify *SROIQ* ontologies. The approach is implemented in the prototype ComR, a *Com*bined *R*easoner. In our approach, we do not need to modify the internals of the existing reasoners and only use them in a black-box manner. Moreover, the proposed decomposition strategy provides

ComR with a more reasonable task assignment. In our approach, the OWL 2 reasoner is responsible for a smaller part of the workload and called *assistant reasoner* (AR); the OWL 2 EL reasoner is responsible for the bulk of the workload and called *main reasoner* (MR). In detail, given an *SROIQ* ontology O, ComR proceeds as per the following two steps:

1) Task decomposition (i) Decomposing the ontology O into an EL subontology O_{EL} and a hybrid subontology O_H, such that O_{EL} is the largest OWL 2 EL ontology and all the non-EL axioms are contained in O_H. Moreover, no logical dependency exists between O_{EL} and O_H, and hence they can be classified in parallel. (ii) Computing a smaller non-EL ontology $M_{\overline{EL}} \subseteq O_H$ (and thus $M_{\overline{EL}} \subseteq O$) such that $M_{\overline{EL}}$ contains only the necessary axioms that can be used to completely classify classes in non-EL axioms.
2) Modular classification Using MR to classify O_{EL}, and combining MR with AR to classify O_H.

The main contributions of this paper can be summarized as follows:

- To optimize the workload, we present an algorithm to compute the largest EL subontology and identify the minimal non-EL subontology.
- We propose a modular reasoning approach that combines a dedicated OWL 2 EL reasoner with a general OWL 2 reasoner for ontology classification, and we prove the correctness of this proposed approach.
- We implement our method in a Java prototype named ComR and provide an extensive experimental evaluation measuring the effectiveness of our techniques.

Our approach is based on the technology of atomic decomposition (AD) [40] that provides a method to understand the topicality and modular structure of an ontology [41, 42]. Through AD, an ontology can be represented and maintained in a decomposed form that contains both axioms and relations over these axioms. The decomposed ontology facilitates the computation of the intended modules and allows us to assemble modules efficiently before reasoning [43]. Hence, our hypothesis is that the ontology is maintained in the decomposed form instead of the traditional way of storing an ontology as a plain collection of axioms. Extensive empirical studies in [44] confirm that our hypothesis is reasonable and practical. We show that an ontology can be decomposed into several independent modules, making possible the concurrent classification of such modules. A key distinguishing feature of our technique is that it can identify the minimal non-EL subontology in an expressive ontology, allowing the delegation of the bulk of the workload to an efficient OWL 2 EL reasoner and the minimal non-EL subontology to an OWL 2 reasoner. In [35], the authors have conjectured that it might be possible to explore the AD techniques to compute a larger EL module for the modular reasoner MORe. On the one hand, our approach confirms this conjecture. On the other hand, our approach shows that the larger EL module possibly leads to unnecessary re-computation in MORe due to its very large size. In addition, our research result provides MORe with some suggestions for optimizing the reasoning task.

The remainder of this paper is structured as follows. Section 2 gives a brief introduction of DL and ontology modularity. Section 3 proposes a task decomposition (TD) algorithm for computing the EL subontology and the non-EL subontology. Section 4 proposes a modular reasoning algorithm for ontology classification and implements it in a prototype ComR. Section 5 evaluates ComR by comparing it with other reasoners. Sections 6 and 7 discuss related works and summarize our work, respectively.

This paper is an extended version of our previous report in the proceedings of ICTAI-13 [45]. Our source codes and test ontologies are available from the project ComR on the platform GitHub.

2 Preliminaries

2.1 The description logic *SROIQ* and EL^{++}

In this paper, we focus on ontologies expressed in OWL 2 [3, 4] and its profile OWL 2 EL [26] which have become the standard ontology languages of W3C. As mentioned in the previous section, the DL *SROIQ* and EL^{++} provide the logical underpinning for OWL 2 and OWL 2 EL, respectively. For convenience, we mainly adopt the DL notion rather than the OWL syntax in definitions and examples.

We refer to individuals, atomic concepts, and atomic roles by calling them *terms*. A set of terms is called a *signature*. A *SROIQ* ontology is based on three finite signature sets of terms: a set N_C of concept names, a set N_R of role names, and a set N_I of individual names. Usually, these sets are assumed to be fixed for some applications and so not mentioned explicitly. The *SROIQ* role expressions are formed according to the following syntax:

$$\mathbf{R} \rightarrow U \mid R \mid R^{-},$$

where U is the universal role and R^- is the inverse of R, $R \in N_R$. Based on this, the *SROIQ* concept expressions are formed according to the following syntax:

$$\mathbf{C} \rightarrow A \mid C \sqcap D \mid C \sqcup D \mid \neg C \mid \top \mid \bot \mid \exists R.C \mid \forall R.C \mid \geqslant nR.C \mid \leqslant nR.C \mid \exists R.Self \mid \{a\},$$

where n is a nonnegative integer and $C, D \in \mathbf{C}$, $A \in N_C$, $a \in N_I$. The top concept $\top$ is always interpreted as the whole domain and the bottom concept $\bot$ is interpreted as the empty set. *Self* is local reflexivity and $\exists R.Self$ represents the set of individuals related to themselves via a given role.

Using the above sets of individual names, role names, and concept names, the axioms of *SROIQ* can be defined as follows:

$$RBox: R \sqsubseteq S, R \equiv S, R \circ S \sqsubseteq P, disjoint(R, S),$$
$$TBox: C \sqsubseteq D, C \equiv D,$$
$$ABox: C(a), R(a, d), a = b, a \neq b,$$

where $a \in N_I$, $b \in N_I$.

The *EL* family of DLs is characterized by allowing unlimited use of existential quantifiers and concept intersection. The original *EL* allows only those features and $\top$. Further extensions of this language are known as EL^+ and EL^{++}. The largest such extension allows the constructors $\sqcap$, $\top$, $\bot$, $\exists$, Self, nominal, and the universal roles, and it supports all types of axioms other than role symmetry, asymmetry, and irreflexivity. The set of EL^{++} concept expressions $\mathbf{C}$ is defined as

$$\mathbf{C} \rightarrow A \mid C \sqcap D \mid \top \mid \bot \mid \exists R.C \mid \exists R.Self \mid \{a\}.$$

The formal semantics of DL axioms is given by their model-theoretic semantics. An *interpretation* I is a pair $I = (\Delta^I, \cdot^I)$, where Δ^I is a non-empty set, called the domain of the interpretation, and $\cdot^I$ is the interpretation function that assigns to each $R \in N_R$ a binary relation $R^I \subseteq \Delta^I \times \Delta^I$, to each $A \in N_C$ a set $A^I \subseteq \Delta^I$, and to each $a \in N_I$ an element $a^I \in \Delta^I$ of the interpretation domain. The interpretation function $\cdot^I$ can be extended to complex roles and concepts according to the corresponding expression. An interpretation I *satisfies* an axiom α (denoted by $I \models \alpha$) if the formula α^I obtained by mapping each term in α via the interpretation function $\cdot^I$ is true. An interpretation I is a *model* of an ontology O if I satisfies each element of O. An ontology O *implies* an axiom α (written $O \models \alpha$) if $I \models \alpha$ for every model I of O. In this case, we say that α is a logical consequence of O. An ontology O is *consistent* if a model of O exists. A concept B is subsumed by C in O if and only if $O \models B \sqsubseteq C$. The computation of the subsumption relationships between all pairs of atomic concepts occurring in O is called *ontology classification*. For a comprehensive understanding of the semantics of DLs, readers are advised to refer to the Description Logic Handbook [17].

2.2 Inference in *SROIQ* and EL^{++}

The reasoning implementations for *SROIQ* are mainly based on the tableau (calculi) [19] or hypertableau (calculi) [20]. Such procedures classify an input ontology, in general, by iterating over all necessary pairs of classes, and trying to build a model of the ontology that violates the subsumption. In the worst case, the computational complexity of this tableau algorithm is N2EXPTIME-complete [18].

In [27], a polynomial-time classification procedure has been proposed for DL EL^{++}. The procedure applies many so-called *completion rules* that derive new concept inclusions. Table 1 presents the completion rules relevant to EL^{++}. These rules are applied to a normalized EL^{++} ontology O that contains only axioms of the form

$$(1)\ A \sqsubseteq B \quad (2)\ A \sqcap B \sqsubseteq C \quad (3)\ A \sqsubseteq \exists R.B$$
$$(4)\ \exists R.A \sqsubseteq B \quad (5)\ R \sqsubseteq S,$$

where A, B, and C are either atomic concepts or $\top$, and R, S are atomic roles.

The completion rules in Table 1 derive new axioms of the form $A \sqsubseteq B$ and $C \sqsubseteq \exists R.D$ from axioms in O and other axioms of these forms already derived, where A, B, C, and D are either atomic concepts or $\top$, and R atomic role. It has been shown that the rules $\mathbf{R}_1$–$\mathbf{R}_7$ are complete for EL^{++} classification [27].

Table 1 Completion rules for EL^{++}

Rule	Description
$\mathbf{R}_1$	If $A \in O$, then $A \sqsubseteq A$
$\mathbf{R}_2$	If $A \in O$, then $A \sqsubseteq \top$
$\mathbf{R}_3$	If $A \sqsubseteq B$ and $B \sqsubseteq C \in O$, then $A \sqsubseteq C$
$\mathbf{R}_4$	If $A \sqsubseteq B$, $A \sqsubseteq C$, and $B \sqcap C \sqsubseteq D \in O$, then $A \sqsubseteq D$
$\mathbf{R}_5$	If $A \sqsubseteq B$ and $B \sqsubseteq \exists R.C \in O$, then $A \sqsubseteq \exists R.C$
$\mathbf{R}_6$	If $A \sqsubseteq \exists R.B$ and $R \sqsubseteq S \in O$, then $A \sqsubseteq \exists S.B$
$\mathbf{R}_7$	If $A \sqsubseteq \exists R.B$, $B \sqsubseteq C$, and $\exists R.C \sqsubseteq D \in O$, then $A \sqsubseteq D$

2.3 Modularity and atomic decomposition

We briefly sketch some central notions around locality-based modularity [46] and AD [40]. We use O to denote an ontology, and M ($M \subseteq O$) for modules. We use $\widetilde{\alpha}$ (resp. $\widetilde{O}$) to denote the signature in an axiom α (resp. O), i.e., the set of concept names, role names, and individual names used in α (resp. O).

A module is a subset of a given ontology that captures all

the knowledge about a specified signature Σ. This intuition is typically formalized by using the notion of conservative extensions (CEs) [46].

Definition 1 Let $M \subseteq O$ be a consistent ontology and $\Sigma \subseteq \widetilde{O}$ a signature.

1) O is a deductive conservative extension (dCE) of M w.r.t. Σ if for all axioms α with $\widetilde{\alpha} \subseteq \Sigma$, it holds that $M \models \alpha$ if and only if $O \models \alpha$.
2) M is a dCE-based module w.r.t. Σ of O if O is a dCE of M w.r.t. Σ.

In other words, M is a dCE-based module w.r.t. Σ of O (denoted by M_Σ) if M and O have the same entailments over Σ. The problem of checking whether M is a module of O w.r.t. Σ is, however, already undecidable for lightweight fragments of OWL 2, such as the OWL 2 EL profile [46]; therefore, approximations are needed in practice. The popular approximation of dCE-based modules is locality-based modules (LBMs), which can be extracted in polynomial time in the size of O [46]. LBMs provide strong logical guarantees: they capture all relevant entailments about Σ. LBM extraction algorithms are implemented in the OWL API on the GitHub site.

Two basic module types of LBMs are $\bot$ and $\top$. Given an ontology O and a seed signature Σ, an axiom $\alpha \in O$ is $\bot$-*local* w.r.t. Σ, if all terms contained in α and not occurring in Σ are replaced with $\bot$, the resulting axiom is a tautology. Then, a $\bot$-module for Σ contains all axioms that are *non*-$\bot$-*local* w.r.t. Σ, plus all those needed to preserve the meaning of terms occurring in these axioms. The $\top$-modules can be defined similarly. Given a module type $x \in \{\top, \bot\}$, we use M^x_Σ to denote the module w.r.t Σ with type x. For intuitive understanding, we present the principle in Example 1. For a formal definition, we refer readers to [46]. Roughly speaking, a $\top$-module for a signature Σ gives a view from "above" because it contains all subclasses of class names in Σ, while a $\bot$-module for Σ gives a view from "below" since it contains all superclasses of class names in Σ. This property of $\bot$-modules can be formalized in the following proposition.

Proposition 1 [46] Let O be an *SROIQ* ontology, A, B concept names in $\widetilde{O} \cup \{\top, \bot\}$, $\Sigma \subseteq \widetilde{O}$ with $A \in \Sigma$, and M a $\bot$-module in O w.r.t. Σ. We conclude that $O \models A \sqsubseteq B$ if and only if $M \models A \sqsubseteq B$.

Proposition 1 shows that, for each class A in a $\bot$-module M, M contains all the superclasses of A. This property makes it well-suited for optimizing ontology classification [35, 47]. In our implementation, we also use a $\bot$-module. The following example shows how to extract a $\bot$-module from an ontology.

Example 1 Consider the ontology Teetotaller containing the following axioms (the non-EL axioms are underlined):

α_1 : $\underline{\text{Animal} \sqsubseteq (= 1\text{hasGender}.\top)}$;
α_2 : $\underline{\text{Animal} \sqsubseteq (\geqslant 1\text{hasHabitat}.\top)}$;
α_3 : Person $\sqsubseteq$ Animal;
α_4 : $\underline{\text{Vegan} \equiv \text{Person} \sqcap \forall \text{eats}.(\text{Vegetable} \sqcup \text{Mushroom})}$;
α_5 : $\underline{\text{TeeTotaller} \equiv \text{Person} \sqcap \forall \text{drinks}.\text{NonAlcoholicThing}}$;
α_6 : Student $\sqsubseteq$ Person $\sqcap \exists$hasHabitat.University;
α_7 : GraduateStudent $\equiv$ Student $\sqcap \exists$hasDegree.BAorBS;
α_8 : University $\sqsubseteq$ EducationOrganization;
α_9 : EducationOrganization $\sqsubseteq$ Organization;
α_{10} : Car $\sqsubseteq$ Vehicle.

Assume that our goal is to extract a $\bot$-module for Σ = {GraduateStudent}. Since the three terms (*Student*, *hasDegree*, *BAorBS*) are contained in α_7 and do not occur in Σ, they are replaced with $\bot$.

$$\alpha_7 : \text{GraduateStudent} \equiv \underbrace{\overbrace{\text{Student}}^{\bot} \sqcap \exists \overbrace{\text{hasDegree}}^{\bot} . \overbrace{\text{BAorBS}}^{\bot}}_{\bot}$$

It is easy to see that the resulting axiom is not a tautology, that is, α_7 is non-$\bot$-local w.r.t. Σ and should be contained in $M^\bot_\Sigma$, i.e., $M^\bot_\Sigma = \{\alpha_7\}$. Now, Σ is extended with new terms (Student, hasDegree, BAorBS), i.e., Σ = {GraduateStudent, Student, hasDegree, BAorBS}. Similarly, we can identify that α_6 is non-local w.r.t the extended Σ and should be contained in $M^\bot_\Sigma$. When no more terms need to be added to the module's signature, we obtain the final extracted module $M^\bot_\Sigma$=$\{\alpha_1, \alpha_2, \alpha_3, \alpha_6, \alpha_7, \alpha_8, \alpha_9\}$.

If the number of terms in an ontology is m, there are 2^m possible signatures to choose from for extracting a module; thus, the number of all modules can be exponential w.r.t. the size of terms. Therefore, it is not feasible to extract all modules. This gives rise to the notion of *genuine modules* [40]:

Definition 2 Given an ontology O, a module $M \subseteq O$ is *fake* if there are (at least) two modules M_1 and M_2 such that $M = M_1 \cup M_2$ with $M_1 \not\subseteq M_2$ and $M_2 \not\subseteq M_1$. A module is called a *genuine module* if it is not fake.

The genuine modules are of interest because they can be used to define a base for all modules; every module can be obtained as the union of suitable genuine modules. Moreover,

genuine modules can be extracted using terms in a single axiom, and the number of genuine modules is linear in the size of an ontology.

During module extraction, it can be observed that different axioms might lead to the same module. For instance, we have $M^{\bot}_{\widetilde{\alpha_1}}$=$M^{\bot}_{\widetilde{\alpha_2}}$ ={α_1, α_2} in Example 1. This reveals a strong logical interrelation between axioms. The set of axioms that "cling together" is formalized using the notion of *atom* [40]:

Definition 3 Given an ontology O, a module type $x \in \{\bot, \top\}$, and an axiom set at={$\alpha_1, \alpha_2, \ldots, \alpha_n$} $\subseteq O$, if $M^x_{\widetilde{\alpha_1}}$= $M^x_{\widetilde{\alpha_2}}$ =, . . . , = $M^x_{\widetilde{\alpha_n}}$, and for any $\beta \in O \setminus at$, $M^x_{\widetilde{\beta}} \neq M^x_{\widetilde{\alpha_i}}$ ($i = 1, 2, \ldots, n$), then at is called an x-atom, denoted by at^x.

In what follows, for simplicity we will omit the atom type x, and say an atom at to mean at^x. Definition 3 shows that, for any module, it either contains all axioms in an atom or none of them. The family of atoms of O is denoted by AD^x_O. Since atoms are pairwise disjoint, AD^x_O is a partition of the ontology, and the number of atoms is linear in the size of the ontology. Given an atom $at \in AD^x_O$, we use M_{at} to denote the module w.r.t the signature $\widetilde{at}$. Definitions 2 and 3 show that M_{at} is a genuine module. Moreover, a 1:1 correspondence exists between atoms and genuine modules. The genuine module for a given atom has the following property:

Proposition 2 [40, Remark 3.10] For each atom at, M_{at} is the smallest module containing at.

The dependent relation among atoms is defined w.r.t the inclusion relation between the corresponding genuine modules:

Definition 4 Let at_1 and at_2 be two atoms in AD^x_O, at_1 is dependent on at_2 (written $at_1 \succeq at_2$) if $M_{at_2} \subseteq M_{at_1}$.

By an abuse of this notation, we say that, for two axiom sets P_1 and P_2 with $P_1 \cap P_2 = \emptyset$, P_1 is dependent on P_2 if there exist atom $at_1 \subseteq P_1$ and $at_2 \subseteq P_2$ such that $at_1 \succeq at_2$. The AD of an ontology O is a pair $(AD^x_O, \succeq)$, denoted by $AD^x_{O,\succeq}$, in which AD^x_O is the set of atoms induced by the genuine modules of O, and $\succeq$ is a partial order (dependency relation) over the set AD^x_O of atoms. The AD is computable in polynomial time and can be visualized by a Hasse diagram, where the independent atoms belong to different chains. The following example shows the AD of an ontology and Fig. 1 presents the corresponding Hasse diagram.

The union of all atoms on which a given atom at depends is called the *principal ideal* of at (denoted by $\overrightarrow{at}$), i.e., $\overrightarrow{at}$=$\bigcup_{at \succeq b} b$.

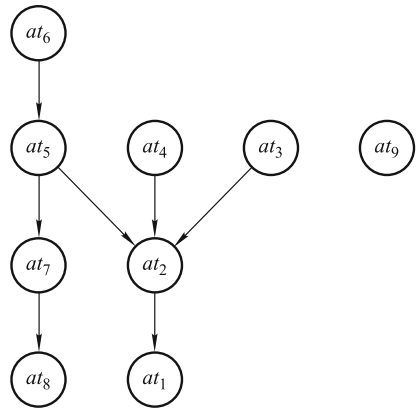


Fig. 1 The Hasse diagram for atomic decomposition of Teetotaller

Proposition 3 [40] Let at be an atom in AD^x_O, the principal ideal $\overrightarrow{at}$ is a module.

Proposition 3 shows that atoms can be seen as building blocks for modules: for each module M, there exist some atoms $at_1, at_2, \ldots, at_k$ in AD^x_O such that M=$\bigcup^k_1 at_i$. The algorithm to perform module extraction from $AD^x_{O,\succeq}$ is described in [48].

Obviously, for each atom at in AD^x_O, $\overrightarrow{at}$ is a genuine module. For a given atom at, the genuine module and the principal ideal coincide:

Proposition 4 Let at be an atom in AD^x_O, $\overrightarrow{at}$=M_{at}

Example 2 Consider AD for the ontology in Example 1. After AD, $AD^{\bot}_{O,\succeq}$ contains nine atoms as follows: at_1 = {α_1, α_2}, at_2 = {α_3}, at_3 = {α_4}, at_4 = {α_5}, at_5 = {α_6}, at_6 = {α_7}, at_7 = {α_8}, at_8 = {α_9}, at_9 = {α_{10}}. The dependency relationships between these atoms are shown in Fig. 1, where the circles represent various atoms and the arrows represent logical dependencies between these atoms, e.g., the arrow from at_2 to at_1 represents $at_2 \succeq at_1$. Since the genuine module and the principal ideal coincide for a given atom, we have M_{at_4}=$\overrightarrow{at_4}$=$at_1 \cup at_2 \cup at_4$ and M_{at_5}=$\overrightarrow{at_5}$=$at_1 \cup at_2 \cup at_5 \cup at_7 \cup at_8$.

Since locality-based modules are defined w.r.t. consistent ontologies and AD is induced by these modules, AD works only for consistent ontologies [49]. In this paper, we consider only consistent ontologies.

3 Task decomposition

The technology of AD shows that an ontology O can be represented in a decomposed form $AD^x_{O,\succeq}$ with $x \in \{\top, \bot\}$ [43]. In this section, we discuss how to identify different modules from such decomposed ontologies so that these modules can

be delegated to appropriate reasoners. We first decompose the ontology O into several independent modules that can be used to construct the EL subontology O_{EL} and the non-EL subontology O_H, and then we identify the minimal non-EL subontology $M_{\overline{EL}}$ from O_H.

3.1 Building independent O_{EL} and O_H

In the Hasse diagram for the AD of an ontology, some nodes with no incoming edges exist, indicating that no other atoms depending on such atoms exist. These atoms are defined as *top atom*.

Definition 5 An atom at_i is *top* if there exists no distinct atom at_j such that $at_j \geq at_i$.

In Example 2, at_3, at_4, at_6, and at_9 are top atoms. It is clear that two distinct top atoms belong to different genuine modules and no dependency relationship exists between them. Moreover, for two different top atoms at_i and at_k, the $\overrightarrow{at_i}$ and $\overrightarrow{at_k}$ are incomparable (w.r.t. set inclusion) modules. In consequence, we have the following lemma:

Lemma 1 Given an ontology O and its AD $AD^x_{O,\geq}$, for each top atom tat in AD^x_O, M_{tat} is the largest genuine module containing tat.

Lemma 1 shows that every ontology can be decomposed into several independent and incomparable (w.r.t. set inclusion) modules if the number of top atoms is greater than one. This can be formalized in the following proposition:

Proposition 5 Let $tat_1, tat_2, \ldots, tat_n$ be all top atoms in $AD^x_{O,\geq}$, the ontology O is equivalent to the union of all the principal ideals of those atoms.

$$O = \bigcup_{k=1}^{n} \overrightarrow{tat_k}. \tag{1}$$

Proof If $tat_1, tat_2, \ldots, tat_n$ be all top atoms in $AD^x_{O,\geq}$, then any atom in AD^x_O is a subset of $\bigcup_{k=1}^{n} \overrightarrow{tat_k}$. Since AD^x_O is a partition of O [40], $\bigcup_{k=1}^{n} \overrightarrow{tat_k}$ contains all logical axioms in O, and this proposition holds. □

Let at be an atom in AD^x_O. at is called a *non-EL atom* if it contains at least one axiom that is outside the OWL 2 EL profile, and at is called an *EL atom* if all the axioms occurring therein fall into the OWL 2 EL profile. In Example 2, there are three non-EL atoms, i.e., at_1, at_3, and at_4. It is noted that the ontology Teetotaller contains four non-EL axioms, i.e., α_1, α_2, α_4, and α_5, which are underlined.

Similarly, we say that a module M is a *non-EL module* if it contains a non-EL axiom, and M is an *EL module* if all axioms occurring therein fall into the OWL 2 EL profile. Since non-EL axioms occur only in non-EL atoms, every non-EL module contains at least one non-EL atom. In contrast, all atoms in an EL module are EL. In what follows, we use S to denote the set of all non-EL atoms in AD^x_O. Given a top atom tat, the module $\overrightarrow{tat}$ is either a non-EL or an EL module determined by whether or not the atom tat depends on some non-EL atom. More precisely, if the tat depends on some non-EL atom in S, then the module $\overrightarrow{tat}$ belongs to O_H; otherwise, the module $\overrightarrow{tat}$ belongs to O_{EL}.

$$O_H = \bigcup_{\substack{\exists b \in S \text{ s.t.} \\ tat \geq b}} \overrightarrow{tat}. \tag{2}$$

$$O_{EL} = \bigcup_{\substack{\nexists b \in S \text{ s.t.} \\ tat \geq b}} \overrightarrow{tat}. \tag{3}$$

Hence Eq. (1) can be represented as follows:

$$O = O_H \bigcup O_{EL}. \tag{4}$$

From Proposition 3 and Lemma 1, O_{EL} is the largest EL subontology that can be classified completely. Although O_{EL} and O_H may share some EL axioms, no logical dependency exists between the two subontologies and they can be concurrently classified by independent reasoners (to be discussed in Section 4).

3.2 Identifying the minimal non-EL subontology $M_{\overline{EL}}$

In this subsection, we discuss how to compute a smaller non-EL ontology $M_{\overline{EL}}$ in O_H such that $M_{\overline{EL}}$ contains only the necessary axioms that can be used to completely classify classes in all non-EL axioms.

Based on the discussion above, it can be observed that, for each top atom tat in O_H, there exist two cases: (i) tat is a non-EL atom, and (ii) tat is an EL atom that depends on some non-EL atom. For the latter case, tat is not necessary for classifying classes in the non-EL atoms on which tat depends and can be ruled out. Indeed, for two distinct atoms at_1 and at_2 with $at_1 \geq at_2$, from Proposition 3, $\overrightarrow{at_2}$ is enough to completely classify classes in the module $\overrightarrow{at_2}$, and at_1 is not necessary. Intuitively, all "upper" atoms are not necessary for classifying classes in the "lower" atoms in the Hasse diagram for AD. Hence, if not all the top atoms in O_H are non-EL, there exists a smaller non-EL subontology $M_{\overline{EL}} \subset O_H$ such that $M_{\overline{EL}}$ is enough to ensure the completeness of classifying classes in non-EL axioms. Each non-EL module can be

assembled using only the "highest" non-EL atom and its dependencies and all the non-EL modules compose the subontology $M_{\overline{EL}}$. For ease of presentation, we define the non-EL subontology $M_{\overline{EL}}$ as follows:

Definition 6 Given the set S of non-EL atoms in AD_O^x, the non-EL subontology $M_{\overline{EL}}$ is equivalent to the union of the principal ideals of all atoms in S.

$$M_{\overline{EL}} = \bigcup_{nat \in S} \overrightarrow{nat}. \tag{5}$$

Proposition 6 Given an $AD_{O,\geq}^x$, $M_{\overline{EL}}$ is the smallest subontology that contains all non-EL atoms and ensures the completeness of classifying classes in non-EL axioms.

Proof For each atom $nat \in S$, according to Propositions 2 and 4, $\overrightarrow{nat}$ is the smallest module including nat. Hence, $M_{\overline{EL}}$ is the smallest subontology including all non-EL atoms in $AD_{O,\geq}^x$. From Proposition 3, $\overrightarrow{nat}$ is a dCE-based module; therefore, $\overrightarrow{nat}$ can be used to completely classify all classes therein. $M_{\overline{EL}}$ is the union of principal ideals of all atoms in S; hence, it can be used to completely classify all classes therein and this proposition holds. □

Equations (2) and (3), and Definition 6 show how to compute O_{EL}, O_H, and $M_{\overline{EL}}$ from a given $AD_{O,\geq}^x$, respectively. Let SN be the set of all non-EL axioms in O. Algorithm 1 is our algorithm for computing O_{EL}, O_H, and $M_{\overline{EL}}$. The following example (Example 2) explains how to divide a decomposed ontology into O_{EL}, O_H, and $M_{\overline{EL}}$.

Example 3 Consider the decomposed ontology Teetotaller in Example 2, where at_3, at_4, at_6, and at_9 are top atoms, and at_1, at_3, and at_4 are non-EL atoms. According to Eq. (1), $O = \overrightarrow{at_3} \cup \overrightarrow{at_4} \cup \overrightarrow{at_6} \cup \overrightarrow{at_9}$. The top atoms all depend on some non-EL atom except at_9; according to Eq. (2), $O_H = \overrightarrow{at_3} \cup \overrightarrow{at_4} \cup \overrightarrow{at_6}$, according to Eq. (3), $O_{EL} = \overrightarrow{at_9}$, and according to Eq. (5), $M_{\overline{EL}} = \overrightarrow{at_1} \cup \overrightarrow{at_3} \cup \overrightarrow{at_4}$.

4 Classification with the combined reasoners

After TD, we can proceed to classify an ontology O using ComR. We denote the classifications of O, O_{EL}, and O_H with C_O, $C_{O_{EL}}$, and C_{O_H}, respectively. Since $O = O_{EL} \bigcup O_H$, we have $C_O = C_{O_{EL}} \bigcup C_{O_H}$. Since no logical dependency relationship exists between O_{EL} and O_H, the two subontologies can be classified in parallel. We classify the EL part O_{EL} using only an OWL 2 EL reasoner (acting as MR). As for O_H, we first use an OWL 2 reasoner (acting as AR) to compute a basic subsumption relationship set $C_{M_{\overline{EL}}}$ from $M_{\overline{EL}}$ such that $C_{M_{\overline{EL}}}$ contains EL^{++} normal forms and subsumption of the form $A \sqcap \exists R.B \sqsubseteq C$ and $\exists R.A \sqcap \exists S.B \sqsubseteq C$, where A, B, and C are atomic concepts, and $RandS$ atomic roles. Then $C_{M_{\overline{EL}}}$ is incorporated into the remaining EL part of O_H to form a temporary EL ontology $O_H^{'}$, i.e., $O_H^{'} = C_{M_{\overline{EL}}} \bigcup (O_H \setminus M_{\overline{EL}})$. Finally, $O_H^{'}$ is delegated to the OWL 2 EL reasoner to obtain the complete classification C_{O_H}. Algorithm 2 describes the entire classification process and Fig. 2 demonstrates the framework and information flow of ComR, where the TD component is

Algorithm 1 Task decomposition

Input: a decomposed ontology $AD_{O,\geq}^{\perp}$
Output: O_{EL}, O_H, $M_{\overline{EL}}$

$O_{EL} \leftarrow \emptyset$, $O_H \leftarrow \emptyset$, $M_{\overline{EL}} \leftarrow \emptyset$
for each $at \in AD_O^{\perp}$ **do**
  if $\nexists a \in AD_O^{\perp}$ such that $a \geq at$ **then**
    $T_{tat} \leftarrow T_{tat} \bigcup \{at\}$ // computing the top atom set: T_{tat}
  end if
  if $at \cap SN \neq \emptyset$ **then**
    $S \leftarrow S \bigcup \{at\}$ // computing the non-EL atom set : S
  end if
end for
for each $tat \in T_{tat}$ **do**
  if $\exists b \in S$ such that $tat \geq b$ **then**
    $O_H \leftarrow O_H \bigcup \overrightarrow{tat}$
  else
    $O_{EL} \leftarrow O_{EL} \bigcup \overrightarrow{tat}$
  end if
end for
for each $nat \in S$ **do**
  if $nat \notin M_{\overline{EL}}$ **then**
    $M_{\overline{EL}} \leftarrow M_{\overline{EL}} \bigcup \overrightarrow{nat}$
  end if
end for
return O_{EL}, O_H, $M_{\overline{EL}}$

Algorithm 2 Classify with ComR

Input: a decomposed OWL 2 ontology O
Output: the classification C_O

$O_H, O_{EL}, M_{\overline{EL}} \leftarrow$ task-Decomposition //Algorithm 1
if $O_{EL} \neq \emptyset$ **then**
  $C_{O_{EL}} \leftarrow MR.classify(O_{EL})$
end if
if $O_H = M_{\overline{EL}}$ **then**
  $C_{O_H} \leftarrow AR.computeSubsumption(O_H)$
else
  $C_{M_{\overline{EL}}} \leftarrow AR.computeSubsumption(M_{\overline{EL}})$
  $O_H^{'} \leftarrow C_{M_{\overline{EL}}} \bigcup (O_H \setminus M_{\overline{EL}})$
  $C_{O_H} \leftarrow MR.classify(O_H^{'})$ // OWL 2 EL reasoner
end if
$C_O \leftarrow C_{O_{EL}} \bigcup C_{O_H}$
return C_O

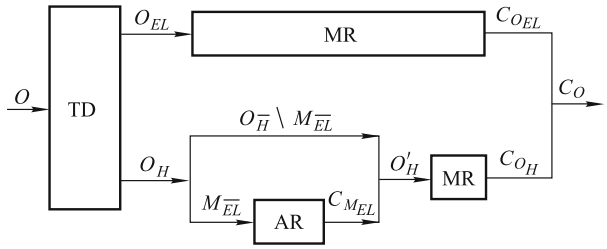


Fig. 2 Information flow of ComR

used to compute the subontologies O_{EL}, O_H, and the minimal non-EL ontology $M_{\overline{EL}}$.

Next, we introduce the following important theorem that ensures the correctness of Algorithm 2.

Theorem 1 Let O be an ontology. Let $C_{O_{EL}}$ and C_{O_H} be as computed by Algorithm 2. Then, for any two atomic concepts $A, B \in O$, $O \models A \sqsubseteq B$ if and only if $C_{O_{EL}} \bigcup C_{O_H} \models A \sqsubseteq B$.

Proof According to Eqs. (1) and (4), the original ontology O is the union of O_{EL} and O_H, and no logical dependency exists between O_{EL} and O_H. $C_{O_{EL}}$ and C_{O_H} encode the classification of O_{EL} and that of O_H, respectively. Hence, the classification of O is the union of the classifications of O_{EL} and O_H, i.e., $C_O = C_{O_{EL}} \bigcup C_{O_H}$, $O \models A \sqsubseteq B$ if and only if $C_{O_{EL}} \bigcup C_{O_H} \models A \sqsubseteq B$.

Clearly, O_{EL} can be classified correctly by an OWL 2 EL reasoner. It remains to show the correctness of the classification of O_H. Let $tat_1, tat_2, \ldots, tat_m$ be all top atoms in O_H; from Eq. (2), $O_H = \bigcup_{j=1}^{m} \overrightarrow{tat_j}$. For two distinct top atoms tat_j and tat_k in O_H, the $\overrightarrow{tat_j}$ and $\overrightarrow{tat_k}$ are two independent genuine modules; hence, the classification of O_H is the union of all the classifications of $\overrightarrow{tat_j}$ ($j = 1, 2, \ldots, m$). Without loss of generality, we consider only one genuine module for a top atom $tat \in O_H$. If tat is a non-EL atom, the module $\overrightarrow{tat}$ is already contained in $M_{\overline{EL}}$ and classified by an OWL 2 reasoner. If tat is an EL atom and there exists a non-EL atom nat with $tat \succeq nat$, it is slightly complicated to classify $\overrightarrow{tat}$ since both the OWL 2 and OWL 2 EL reasoners are involved. Let C be a class in $\overrightarrow{tat}$. We use $S(C)$ to denote the superclass of C and consider the following two cases:

Case 1 C is in the module $\overrightarrow{nat}$. Then, from Proposition 1, the module $\overrightarrow{nat}$ contains all superclasses of C and those superclasses have already been computed by the OWL 2 reasoner.

Case 2 C is not in the module $\overrightarrow{nat}$, that is, C occurs in the remaining EL part $\overrightarrow{tat} \setminus \overrightarrow{nat}$. Then, the $S(C)$ is possibly in $\overrightarrow{nat}$ or $\overrightarrow{tat} \setminus \overrightarrow{nat}$. If $S(C)$ is in the module $\overrightarrow{nat}$, then $S(C)$ is computed by the EL completion rules using normal forms from $C_{M_{\overline{EL}}}$ and $\overrightarrow{tat} \setminus \overrightarrow{nat}$. If $S(C)$ is in $\overrightarrow{tat} \setminus \overrightarrow{nat}$, then $S(C)$ is computed using only those normal forms from $\overrightarrow{tat} \setminus \overrightarrow{nat}$. □

To conclude, we illustrate the process of Algorithm 2 by the following example.

Example 4 Consider the ontology Teetotaller in Example 2, where $O_H = \overrightarrow{at_3} \cup \overrightarrow{at_4} \cup \overrightarrow{at_6} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9\}$, $O_{EL} = \overrightarrow{at_9} = \{\alpha_{10}\}$, and $M_{\overline{EL}} = \overrightarrow{at_1} \cup \overrightarrow{at_3} \cup \overrightarrow{at_4} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$.

The classification of the subontology O_{EL} leads to the following subsumption $C_{O_{EL}}$: Car $\sqsubseteq$ Vehicle.

The $C_{M_{\overline{EL}}}$ computed from $M_{\overline{EL}}$ contains the following subsumption relationships

$$\begin{aligned} \text{Person} &\sqsubseteq \text{Animal},\\ \text{Vegan} &\sqsubseteq \text{Person},\\ \text{Vegan} &\sqsubseteq \text{Animal},\\ \text{TeeTotaller} &\sqsubseteq \text{Person},\\ \text{TeeTotaller} &\sqsubseteq \text{Animal}. \end{aligned}$$

The normalized forms of the remaining EL part (α_6, α_7, α_8, α_9), denoted by NF_{EL}, contains the following subsumption relationships:

$$\begin{aligned} \text{Student} &\sqsubseteq \text{Person},\\ \text{Student} &\sqsubseteq \exists \text{hasHabitat.University},\\ \text{GraduatedStudent} &\sqsubseteq \text{Student},\\ \text{GraduatedStudent} &\sqsubseteq \exists \text{hasDegree.BAorBS},\\ \text{Student} \sqcap \text{AUX} &\sqsubseteq \text{GraduatedStudent},\\ \exists \text{hasDegree.BAorBS} &\sqsubseteq \text{AUX},\\ \text{University} &\sqsubseteq \text{EducationOrganization},\\ \text{EducationOrganization} &\sqsubseteq \text{Organization}, \end{aligned}$$

where AUX is a new atomic concept introduced in the process of normalization.

Next, $C_{M_{\overline{EL}}}$ is incorporated into the remaining EL part $O_H \setminus M_{\overline{EL}}$ to obtain all the remaining atomic subsumption relationships that hold in O_H.

$$\begin{aligned} \text{GraduatedStudent} &\sqsubseteq \text{Person},\\ \text{Student} &\sqsubseteq \text{Animal},\\ \text{GraduatedStudent} &\sqsubseteq \text{Animal}, \end{aligned}$$

As shown above, the four atomic concepts (Animal, Person, Vegan, and TeeTotaller) occur in $M_{\overline{EL}}$ and their superclasses are computed by an OWL 2 reasoner. The two atomic concepts (Student and GraduatedStudent) are not in $M_{\overline{EL}}$ and

their superclasses are computed by the *EL* completion rules using axioms in NF_{EL} and $C_{M_{\overline{EL}}}$.

5 Empirical evaluation

To evaluate the effectiveness of our techniques in ontology classification, in this section we present an empirical evaluation of the proposed modular reasoning techniques described over a collection of commonly used ontologies. In Section 5.1, we describe our test ontologies and experimental setup. In Section 5.2 we first evaluate our TD algorithm from Section 3, and then we evaluate our combined reasoner ComR from Section 4 and compare it with HermiT, FaCT++, Pellet, Konclude, and MORe. Section 5.3 is devoted to the comparison between MORe and ComR since MORe is the closest approach.

5.1 Test ontologies and experimental environment

We have implemented ComR in Java using the OWL API. We use two Java threads to perform the classifications of O_{EL} and O_H in parallel if both O_{EL} and O_H are not empty. The implementation of Algorithm 1 is based on $AD^{\perp}$. In ComR, ELK 0.4.1 is chosen as the main reasoner (MR) and the state-of-the-art HermiT 1.3.8 acts as the assistant reasoner (AR). In our experiment, we use HermiT 1.3.8, FaCT++ 1.6.3, Pellet 2.3.1, Konclude 0.6.2, and MORe in combination ELK 0.4.1 with HermiT 1.3.8. All the reasoners are available from the website of University of Manchester.

Our test ontology suite contains 23 ontologies available from the NCBO BioPortal ontology repository and the TONES ontology repository. The majority of test ontologies are also used in evaluations of other works such as [21], [35], [38], and [50], which enable a concrete comparison to directly show the effect of our approach for well-known benchmark ontologies. Since our approach is proposed for expressive OWL 2 ontologies, all test ontologies contain non-EL axioms. Table 1 shows the statistics of metrics for each ontology, including the language in which they are expressed, the number of atomic concepts (classes), roles, axioms, and non-EL axioms they contain. It is noted that only the logical axioms are counted.

All experiments were conducted on a laptop with an AMD quad core A6-3400 APU and 16GB of memory running Windows 7. We ran Java 1.7 with the -XX:+AggressiveHeap option to use all available physical memory.

5.2 Evaluations

We have run Algorithm 1 for TD and Algorithm 2 for classi-

Table 2 Statistics of ontology metrics for the test ontologies

Ontology	C	R	Size	nonEA	DL
TAO	3,371	24	5,188	2	ALERI
OAE	4,673	125	4,388	346	ALCH
SYN	14,462	2	15,353	4	ALF
ERO	4,061	193	5,022	105	SHOIF(D)
Hupson	2,919	91	3,771	62	SHOIF(D)
BCGO	2,269	106	4,514	112	SROIN(D)
CSEO	20,085	91	26,540	14	ALCH
Idoden	5,035	22	5,892	5	SRIF
EDAM	3,220	12	4,360	8	ALCHI
Galen	23,141	950	37,696	1,149	SHIF
Dermlex	6,106	18	24,452	16	ALCHF(D)
OMIT	2,226	126	7476	18	SRI(D)
Protein	35,470	9	46,698	16	SH
NCI	93,413	52	219,224	65	SH(D)
ICNP	4,212	43	11,901	38	SHIF(D)
UBERON	18,874	189	48,806	1,296	SRIQ
EFO	17,892	35	25,173	32	ALHI
ECA	10,030	61	12,243	36	SRIQ(D)
SO	2,311	50	3,035	4	SHI
ENVO	2,250	110	3,001	3	SHI(D)
Amino-acid	46	5	465	111	ALCF(D)
Unit	12	3	358	6	ALUOF(D)
Family-tree	22	52	1,609	31	SROIF(D)

Note: The number of classes (C), the number of roles (R), the number of axioms (Size), the number of non-EL axioms (nonEA), and the DL expressivity (DL)

Table 3 Task decomposition in ComR

Ontology	Size	O_{EL}	$M_{\overline{EL}}$	$O_H \setminus M_{\overline{EL}}$
TAO	5,188	5,178	13	0
OAE	4,388	3,496	892	0
SYN	15,353	15,352	27	0
ERO	5,022	0	508	4,514
Hupson	3,771	0	453	3,318
BCGO	4,514	0	1225	3289
CSEO	26,540	18,517	94	7,929
Idoden	5,892	1,672	7	4,213
EDAM	4,360	1,021	34	3,305
Galen	37,696	1,762	2,165	33,769
Dermlex	24,452	6,105	8,219	10,128
OMIT	7476	2958	562	3,956
Protein	46,698	43,749	288	2,779
NCI	219,224	189,326	7,151	24,278
ICNP	11,901	8,951	682	2,797
UBERON	48,806	36,192	12,988	1,316
EFO	25,173	18,652	897	6,413
ECA	12,243	1,282	277	10,733
SO	3,035	3,001	7	28
ENVO	3,001	2,987	11	4
Amino-acid	465	0	465	0
Unit	358	0	358	0
Family-tree	1,609	0	1,609	0

Table 4 Comparison of ComR with other reasoners on classification (time in s)

Ontology	HermiT	FaCT++	Pellet	Konclude	MORe	ComR		
						T_{total}/s	T_{Cla}/s	T_{TD}/s
TAO	10.72	8.24	14.85	5.26	7.23	2.45	2.41	0.04
OAE	9.43	10.67	11.54	3.27	6.32	1.78	1.54	0.21
SYN	24.82	17.46	27.44	12.83	12.65	5.27	4.34	0.72
ERO	11.62	13.24	10.63	1.28	12.83	2.79	2.02	0.53
HUPSON	8.72	7.43	6.85	1.43	9.24	1.64	1.61	0.02
BCGO	13.72	16.68	17.34	4.37	14.34	5.26	4.82	0.31
CSEO	52.37	37.24	64.75	5.31	20.18	4.83	4.14	0.57
IDODEN	3.23	3.82	4.66	2.43	2.95	2.18	2.04	0.09
EDAM	2.14	3.26	4.28	0.35	2.81	0.56	0.51	0.04
Galen	44.83	29.37	34.74	1.52	25.71	3.84	3.16	0.58
DERMLEX	42.63	34.68	45.82	1.94	16.32	3.26	3.02	0.23
OMIT	4.42	7.26	3.61	2.18	2.24	1.24	1.04	0.08
Protein	47.24	33.62	51.31	3.37	12.85	1.57	1.15	0.41
NCI	518.82	447.46	531.63	12.25	102.37	16.75	15.24	1.53
ICNP	7.62	19.43	8.38	0.79	3.24	0.84	0.78	0.05
UBERON	163.25	132.71	184.54	21.46	78.63	32.84	31.24	1.47
EFO	97.67	104.34	132.38	28.32	64.84	22.87	21.08	1.24
ECA	82.73	65.35	45.16	19.74	32.75	25.46	24.04	1.26
SO	3.58	2.54	1.96	1.82	1.57	1.28	1.26	0.11
ENVO	2.83	4.17	3.72	1.65	1.47	1.14	1.08	0.03
Amino-acid	0.38	0.24	0.34	0.26	0.53	0.46	0.41	0.03
Unit	0.26	0.24	0.23	0.21	0.45	0.39	0.35	0.02
Family-tree	61.63	56.32	58.74	54.56	67.34	64.23	63.19	0.74

fication over the test ontologies. The experimental results are summarized in Tables 3 and 4, respectively.

Table 3 presents the TD in ComR. The third column is the size of the subontology O_{EL}, the fourth column presents the size of subontology $M_{\overline{EL}}$, and the final column provides the number of axioms in the remaining EL part $O_H \backslash M_{\overline{EL}}$, where $M_{\overline{EL}}$ and O_{EL} can be used for complete classification. However, those axioms in the $O_H \backslash M_{\overline{EL}}$ cannot be used for complete classification since they depend on the ones in the $M_{\overline{EL}}$. Table 3 also shows that, for most large ontologies, it is possible to identify a smaller non-EL subontology and delegate it to an OWL 2 reasoner. It is noted that, in some cases, the sum of the two sizes of O_{EL} and O_H is bigger than the size of ontology (the second column) because O_{EL} and O_H share some EL axioms, i.e., there exist some EL axioms that are contained in both $M_{\overline{EL}}$ and O_{EL}. For ease of discussion, we identified four groups of the test ontologies, which we delineate in Table 3 using horizontal lines.

The first group contains three ontologies in which the remaining EL part in O_H is empty, i.e., $O_H = M_{\overline{EL}}$, and then $O_H \setminus M_{\overline{EL}}=\emptyset$. From the point of view of AD, every top atom in O_H is a non-EL atom. In this case, the whole subontology O_H is delegated to an OWL 2 reasoner in ComR.

The second group contains three ontologies in which the EL subontology O_{EL} is empty and $O_H \setminus M_{\overline{EL}}$ is not empty. This means that every top atom depends on some non-EL atom in the AD. In ComR, those ontologies are sequentially classified as shown in our previous work [45], that is, $M_{\overline{EL}}$ is first classified by an OWL 2 reasoner to obtain the basic subsumption set $C_{M_{\overline{EL}}}$, and then $C_{M_{\overline{EL}}} \bigcup (O_H \setminus M_{\overline{EL}})$ is delegated to an OWL 2 EL reasoner.

The third group contains fourteen ontologies in which both $O_H \setminus M_{\overline{EL}}$ and O_{EL} are not empty. In addition, it can be seen that $O_{EL} \bigcap O_H$ is empty for the first six ontologies (CSEO, Idoden, EDAM, Galen, Dermlex, and OMIT), whereas, $O_{EL} \bigcap O_H$ is not empty for the second eight ontologies (Protein, NCI, ICNP, UBERON, EFO, ECA, SO, and ENVO). This means that these shared axioms will be computed twice.

The last group contains three ontologies in which both $O_H \setminus M_{\overline{EL}}$ and O_{EL} are empty. From the point of view of AD, every top atom is a non-EL atom. This means that there exist tight logical interrelations between atoms and each non-EL atom will be involved in a larger module. In this case, the whole ontology O is delegated to an OWL 2 reasoner in ComR.

Table 4 presents the comparison of ComR with other reasoners in terms of classification time. Since the four reason-

ers (HermiT, FaCT++, Pellet, and MORe) perform some preprocessing and Konclude only performs preprocessing on demand during the process of loading, for a fair comparison, the shown times do not include loading time. We use an experimental methodology similar to that in the work of [50] in which the loading time is not included and only the classification time is measured. We access other reasoners through the OWL API 3.4.3 and measure the time spent inside the classification method. It is important to notice that the classification time of MORe includes the module extraction time. For ComR, the total time T_{total} includes the classification time T_{Cla} and the TD time T_{TD}. All figures shown in Table 4 were obtained as the average over three runs of the experiments.

For the first twenty test ontologies, Table 4 shows that our approach outperforms the hypertableau-based reasoner (HermiT) and the tableau-based reasoners (FaCT++, Pellet). Especially in the case that O_{EL} is not empty, O_{EL} and O_H are classified concurrently, leading to a reasonable task assignment and offering a substantial speedup. For example, the size of O_{EL} in NCI is 189326 (86.4%), and the reasoning time is reduced by 96.9% compared against Pellet. For ontologies (the second group) in which O_{EL} is empty, ComR runs sequentially. First, the subsumption relationship set $C_{M_{\overline{EL}}}$ is computed by the OWL 2 reasoner, and then $C_{M_{\overline{EL}}}$ is incorporated into the remaining EL part so as to use it with a dedicated OWL 2 EL reasoner. This procedure is just like the one presented in our previous work [45]. It is easy to see that the smaller $M_{\overline{EL}}$ is, the less time it takes to classify the whole ontology. The ontology ERO contains 105 non-EL axioms and the size (508) of the corresponding $M_{\overline{EL}}$ is very small (10.1%); hence, the bulk of the workload is assigned to the EL reasoner, and ComR obtains a slightly high performance. BCGO, however, contains only 112 axioms that are outside the OWL 2 EL, the size (1225) of $M_{\overline{EL}}$ is slightly larger (27.1%), and a higher workload is assigned to the OWL 2 reasoner, leading to a modest performance gain.

For the last three ontologies (Amino-acid, Unit, and Family-tree), ComR is slower than HermiT, FaCT++, and Pellet. The reason is that both O_{EL} and $O_H \setminus M_{\overline{EL}}$ are empty in those ontologies, and consequently ComR delegates the whole ontology to an OWL 2 reasoner. Moreover, ComR has to spend some time in TD before classification. For the same reason, ComR will be slower than an OWL 2 EL reasoner if the test ontologies do not contain non-EL axioms. In the case that both O_{EL} and $O_H \setminus M_{\overline{EL}}$ are empty, both ComR and MORe will delegate the whole ontology to an OWL 2 reasoner and both of them spend time in identifying ontology modules before classification. However, ComR introduces less overhead than MORe. The reason is that the decomposition technology used by ComR provides a fast way to extract locality-based modules [40]; in contrast, MORe is based directly on the traditional locality-based module extraction algorithm [46].

Although ComR is not universally better than Konclude on all test ontologies, our results indicate that ComR may be slightly efficient on some ontologies in which O_{EL} is large and $M_{\overline{EL}}$ is small, such as TAO and SYS. For the ontologies that contain a larger $M_{\overline{EL}}$, such as NCI and UBERON, ComR is less efficient than Konclude. The possible reasons are (i) Konclude is implemented in C++ and has less overhead than Java implementations; in addition, (ii) Konclude uses more sophisticated optimization techniques, whereas ComR is a prototype.

To sum up, the results from the above experiments suggest that the benefits achieved from combining the EL reasoner with the OWL 2 reasoner include two aspects. First, the technology of AD provides a fast module extraction method, making it is possible to assemble modules quickly and introducing less overhead in the procedure of modular reasoning. Second, ComR provides a more reasonable strategy for task assignment; the bulk of the workload is delegated to the efficient EL reasoner and very low workload is assigned to the less efficient OWL 2 reasoner. Hence, the combined reasoner ComR is especially useful in the following situations:

1) The ontology is large and contains a small number of non-EL axioms.
2) The ontology contains loose dependent relationship between atoms.

5.3 Comparison with MORe

Since MORe [35] is the closest approach, in this subsection, we discuss the comparison between MORe and ComR in detail. From Table 4, ComR is universally better than MORe on all test ontologies. Even in the case that both O_{EL} and $O_H \setminus M_{\overline{EL}}$ are empty, both ComR and MORe delegate the whole ontology to an OWL 2 reasoner, such as the last three ontologies (Amino-acid, Unit, and Family-tree). However, ComR is slightly faster than MORe, and the main reason is that our approach is based on the technology of AD that can contribute to improving the performance of existing Algorithms employed for locality-based module extraction [40, 51], on which MORe is based. Besides fast module extraction, the other reasons possibly lie in two aspects: (i) execution strategy, and (ii) reasoning task assignment. We

discuss them in detail as follows.

Regarding the execution strategy, MORe invokes the OWL 2 reasoner and OWL 2 EL reasoner sequentially. In detail, MORe first computes an EL signature Σ and the complement signature $\overline{\Sigma}$ such that the EL module w.r.t. Σ (denoted by M_{Σ}^{MORe}) does not contains any non-EL axiom. Secondly, the EL module M_{Σ}^{MORe} and non-EL module (denoted by $M_{\overline{\Sigma}}^{MORe}$) are extracted. While classification, the $M_{\overline{\Sigma}}^{MORe}$ is first delegated to an OWL 2 reasoner, and then both the computed result and M_{Σ}^{MORe} are assigned to an OWL 2 EL reasoner to obtain the complete classification. This execution strategy is just like the one in [45]. In case that O_{EL} is not empty, it inevitably takes more time to invoke the OWL 2 reasoner and the OWL 2 EL reasoner in sequence. In contrast, ComR invokes the two kinds of reasoners concurrently. For example, ComR concurrently classifies O_{EL} and O_H in NCI, and the reasoning time is reduced by 83.7% compared against MORe.

As far as the second aspect is concerned, the reasoning task assignment in MORe is coarser than the one in ComR. Indeed, the M_{Σ}^{MORe} (resp. $M_{\overline{\Sigma}}^{MORe}$) in MORe corresponds to O_{EL} (resp. O_H) in ComR. MORe simply delegates the whole O_H to an OWL 2 reasoner. In comparison, we identify a smaller non-EL subontology $M_{\overline{EL}}$ from O_H that is delegated to the less efficient OWL 2 reasoner. The smaller $M_{\overline{EL}}$ is already enough to classify classes in all non-EL axioms completely, as established by Theorem 1. Table 4 shows the comparison between MORe and ComR on reasoning task assignments. From Table 5, the non-EL subontology ($M_{\overline{EL}}$) in ComR is universally smaller than the one ($M_{\overline{\Sigma}}^{MORe}$) in MORe on all test ontologies.

Indeed, given an ontology O and the module type x, the structure of O is uniquely determined by its AD $AD_{O,\geq}^{x}$, and both O_{EL} and O_H are unique. For MORe, it is optimal if M_{Σ}^{MORe}=O_{EL} and $M_{\overline{\Sigma}}^{MORe}$=$O_H$. According to the heuristics used to compute the signature (hence for the corresponding module) in MORe, $M_{\overline{\Sigma}}^{MORe}$ is always larger than or equal to O_H. However, M_{Σ}^{MORe} possibly occurs in three cases:

Case 1 M_{Σ}^{MORe} is smaller than O_{EL}. In this case, MORe cannot achieve optimal task assignment. Apart from O_H, MORe delegates some axioms in O_{EL} to the OWL 2 reasoner, such as the twelve ontologies in our test suite (TAO, OAE, SYN, CSEO, Protein, NCI, ICNP, UBERON, EFO, ECA, SO, and ENVO).

Table 5 Comparison of MORe with ComR on reasoning task assignments

Ontology	MORe		ComR		
	M_{Σ}^{MORe}	$M_{\overline{\Sigma}}^{MORe}$	O_{EL}	$M_{\overline{EL}}$	$O_H \setminus M_{\overline{EL}}$
TAO	5,023	179 (3.5%)	5,178	13 (0.3%)	0
OAE	3,427	985 (22.4%)	3,496	892 (20.3%)	0
SYN	15,294	187 (1.2%)	15,352	27 (0.8%)	0
ERO	836	5,022 (100%)	0	508 (10.1%)	4,514
Hupson	693	3,771 (100%)	0	453 (12.0%)	3,318
BCGO	701	4,514 (100%)	0	1,225 (27.1%)	3,289
CSEO	17,937	8,693 (32.8%)	18,517	94 (0.4%)	7,929
IDODEN	1,753	4,316 (73.3%)	1,672	7 (0.1%)	4,213
EDAM	1,021	3,339 (76.6%)	1,021	34 (0.8%)	3,305
Galen	2,164	35,976 (95.4%)	1,762	2,165 (5.7%)	33,769
Dermlex	6,105	18,347 (75.0%)	6,105	8,219 (33.6%)	10,128
OMIT	3,140	4,703 (62.9%)	2,958	562 (7.5%)	3,956
Protein	42,915	3,972 (8.5%)	43,549	288 (0.6%)	2,779
NCI	186,334	33,760 (15.4%)	189,326	7,151 (3.3%)	24,278
ICNP	8,951	4,381 (36.8%)	9,731	682 (5.7%)	2,797
UBERON	34,984	14,906 (30.5%)	36,192	12,988 (26.6%)	1,316
EFO	18,372	7,348 (29.2%)	18,652	897 (3.6%)	6,413
ECA	1,117	11,292 (92.2%)	1,282	277 (2.3%)	10,733
SO	2,918	153 (5.0%)	3,001	7 (0.02%)	28
ENVO	2,911	104 (3.5%)	2,987	11 (0.4%)	4
Amino-acid	164	465 (100%)	0	465 (100%)	0
Unit	227	358 (100%)	0	358 (100%)	0
Family-tree	1,263	1,609 (100%)	0	1,609 (100%)	0

Note: $M_{\overline{\Sigma}}^{MORe}$ is delegated to the OWL 2 reasoner in MORe, and $M_{\overline{EL}}$ is delegated to the OWL 2 reasoner in ComR

Case 2 M_{Σ}^{MORe} is equal to O_{EL}. In this case, MORe is optimal for task assignment. The case occurs in the two ontologies (EDAM and Dermlex) in our test.

Case 3 M_{Σ}^{MORe} is larger than O_{EL}. In this case, MORe causes over-optimizing in task assignment. M_{Σ}^{MORe} contains some "unnecessary" EL axioms that lead to unnecessary re-computation. These "unnecessary" axioms occur in $O_H \backslash M_{\overline{EL}}$ and they do not depend on any non-EL atoms. It is noted that the shared EL axioms in ComR only occur in $M_{\overline{EL}} \bigcap O_{EL}$, if such axioms exist. For instance, in our example, α_8 and α_9 are "unnecessary" axioms if they belong to M_{Σ}^{MORe}. For the six ontologies (ERO, Hupson, BCGO, IDODEN, Galen, and OMIT), all the M_{Σ}^{MORe} modules contain "unnecessary" axioms. The six ontologies (ERO, Hupson, BCGO, Amino-acid, Unit, and Family-tree) are extreme cases, and all the axioms in M_{Σ}^{MORe} are "unnecessary" because there exists no EL top atom that depends only on EL atoms in the AD. Moreover, all the top atoms in the three ontologies (Amino-acid, Unit, and Family-tree) are non-EL atoms.

For intuition, we use the following example (Example 5) to illustrate the over-optimization in MORe.

Example 5 Consider an ontology O containing the first nine axioms in Teetotaller presented in Example 1, i.e., $O = \{\alpha_1, \alpha_2, \ldots, \alpha_9\}$. Although O_{EL} is empty, ComR still identifies a smaller non-EL subontology $M_{\overline{EL}}$ that is delegated to the OWL 2 reasoner, where $M_{\overline{EL}}$=$\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$. On the contrary, MORe first computes the largest EL signature Σ={University, EducationOrganization, Organization} and extracts the EL module M_{Σ}^{MORe}=$\{\alpha_8, \alpha_9\}$. Then, MORe uses the complement signature $\overline{\Sigma}$=$\widetilde{O} \setminus \Sigma$ to extract the corresponding non-EL module. Since both α_8 and α_9 are not local w.r.t. $\overline{\Sigma}$, the two axioms are contained in the non-EL module, i.e., $M_{\overline{\Sigma}}^{MORe}$=$\{\alpha_1, \alpha_2, \ldots, \alpha_9\}$, which contains the whole ontology O. According to the strategy of MORe, the module $M_{\overline{\Sigma}}^{MORe}$ is first delegated to an OWL 2 reasoner, and the computed result together with M_{Σ}^{MORe} are assigned to an EL reasoner. Consequently, the re-computation of α_8 and α_9 is unnecessary.

Indeed, in the case that O_{EL} is empty, such as for the six ontologies (ERO, Hupson, BCGO, Amino-acid, Unit, and Family-tree), MORe delegates the whole ontology to an OWL 2 reasoner and the special reasoner is unnecessary. This is also confirmed by the comparison between HermiT and MORe on the six ontologies shown in Table 3, where HermiT outperforms MORe. In this test, we use MORe in combination with ELK 0.4.1 and HermiT 1.3.8. In contrast, ComR delegates the whole ontology to an OWL 2 reasoner only if O_{EL} is empty and O_H is equivalent to $M_{\overline{EL}}$, such as the three ontologies (Amino-acid, Unit, and Family-tree).

6 Related work

Recently, some attention is paid to new approaches that combine special reasoning procedures with full-fledged tableau algorithms to improve the reasoning performance for ontologies in expressive DLs.

From the literature, the closest approach is MORe [35], in which an OWL 2 reasoner and an OWL 2 EL reasoner are combined to classify an OWL 2 ontology. This combined reasoner is directly based on module extraction [46]. More precisely, the combined reasoner exploits the property of $\bot$-modules presented in Proposition 1 to optimize classification. Since the $\bot$-modules w.r.t. a signature Σ contain all the superclasses of each class occurring in Σ, MORe uses a heuristics approach to compute a larger EL signature Σ such that the module M_{Σ}^{MORe} only contains EL axioms so as to delegate it to an OWL 2 EL reasoner. However, as mentioned in the previous section, MORe cannot guarantee to obtain an optimal EL signature; hence, the non-EL module $M_{\overline{\Sigma}}^{MORe}$ is not necessarily minimal. In contrast, our approach computes the exact EL subontology whose reasoning can be delegated to a separate and independent OWL 2 EL reasoner. Even in the case that the O_{EL} is empty, our approach still identifies the minimal non-EL subontology whose reasoning is delegated to an OWL 2 reasoner. However, in such a case, MORe delegates the whole ontology to the OWL 2 reasoner. From the point of view of optimization, our method leads to a more reasonable reasoning assignment. In addition, our approach suggests some methods for optimizing MORe: when extracting the non-EL module $M_{\overline{\Sigma}}^{MORe}$, if $M_{\Sigma}^{MORe} \subseteq M_{\overline{\Sigma}}^{MORe}$, the procedure of module extraction can be stopped. In such a case, the whole ontology can be delegated to the OWL 2 reasoner and invoking the OWL 2 EL reasoner is not necessary. The MORe system is directly based on module extraction and it is suitable for original ontologies, which are stored as plain collections of axioms (or triples) in traditional ways. In contrast, our work is suitable for decomposed ontologies that facilitate module extraction and is considered as a desirable way to manage large ontologies [43, 48].

Chainsaw [36] is designed as a wrapper for reasoners. It uses reasoner factories to build reasoners on ontology mod-

ules and chooses the best suited reasoner for each reasoning task on the basis of module characteristics, such as size and/or expressivity. Chainsaw is based on labeled AD [48], which extends AD with some labels that reveal the relations between modules and signatures. Chainsaw focuses on answering user queries over complex OWL ontologies. For every query, the module is constructed: via modularization algorithm for entailment queries and via labeled AD for hierarchical queries. A suitable reasoner is then created for that module, and the query is delegated to it. Finally, the answer is returned to the user. More specifically, the strategy for answering a query Q consists of three steps: (i) selecting a module M for Q, (ii) starting a new reasoner R on M, and (iii) answering the original query Q using R. In labeled AD, many terms are required to label atoms or modules; such additional data for labels consume memory and place an extra burden on the preprocessing of an ontology. Additionally, the use of separate and independent reasoners suggests the possibility of using multiple threads. This feature, however, is not implemented in Chainsaw yet [49]. In contrast, ComR has used multiple threads to take advantage of multiple processors/cores, which increasingly prevail in computer systems.

In [37], Song et al. presented a weakening and strengthening approach for *ALCHO* ontology classification, using a hybrid of consequence-based and tableau-based reasoning approaches. The original ontology O_o is approximated by a weakened version O_w and a strengthened version O_s, and both are in a less expressive DL *ALCH* and can be classified by a consequence-based reasoner. The classification of O_w is sound but possibly incomplete with respect to O_o, while that of O_s is complete but possibly unsound. Therefore, the tableau-based reasoner is required to verify the unsound subsumption derived from O_s. This approach is based on language weakening [52], the ideal of which is based on the well-known trade-off between the expressiveness and the reasoning complexity of logical language. Our work differs from this in that we target at the more expressive DL *SROIQ* and we do not need to encode the original ontology into the other one.

In Konclude [38], a completion-based saturation procedure is coupled with a tableau algorithm for the DL *ALCHOIQ*. To obtain compatible data structures on which the saturation and the tableau algorithm operate, a completion-based saturation procedure is adapted to generate a saturation graph in which the nodes are labeled with sets of concepts and the edges are labeled with sets of roles. Consequently, the completion-based saturation method allows transferring results from the saturation to the tableau algorithm. Compared with MORe, Konclude works more from the opposite direction: it applies the saturation and simply ignores features that are not supported and then detects the parts that cannot be completely handled during reasoning; finally, the unhanded parts are delegated to the tableau algorithm. Our approach is a trade-off between MORe and Konclude: we first compute as large an EL part as possible and delegate it to an efficient OWL 2 EL reasoner, and then we identify the minimal non-EL part from the remainder of the ontology whose reasoning is delegated to a less efficient OWL 2 reasoner. Konclude has to modify the completion-based saturation and tableau-based algorithms and implement them into a reasoning system, whereas in our approach, the existing reasoners need not be modified and only used in a black-box modular manner. In this sense, our approach is much more general and flexible; hence, it is easy to extend with other profile-specific reasoners, such as the reasoner CB for OWL 2 RL [53], the reasoner QuOnto for OWL 2 QL [54], and the reasoner RORS for OWL rule-based parallel reasoning [55,56]. In the case of Konclude, the whole ontology has to be loaded, whereas our approach allows us to load an ontology partially due to the decomposed form.

7 Conclusions and future work

In this paper, we presented a technique that combines an OWL 2 EL with an OWL 2 reasoner for *SROIQ* ontology classification. The combination is based on a suitable task assignment; the bulk of the workload is delegated to an efficient OWL 2 EL reasoner and only the minimal non-EL subontology is delegated to a less efficient OWL 2 reasoner. This approach allows users to use expressive DLs to build their ontologies and still enjoy efficient services as in tractable languages. We demonstrate the performance gain of reasoning indicated by the significant reduction of classification time.

The increasingly large-scale ontologies make it more difficult to manage, maintain, and reuse them. In [48], the large-scale investigation shows a good decomposability of ontologies. For the majority of ontologies in the NCBO BioPortal repository, the average atom in $AD^{\perp}$ is only slightly larger than two axioms. The empirical results of [44] shows that computing ADs is practical. In [43], an approach has been proposed to maintain ontologies in the decomposed form. We believe that our approach provides an effective way to classify such decomposed ontologies.

We intend to continue our work in three directions. First, we will optimize ComR and extend it to handle the other two OWL 2 profiles. Second, it would be interesting to ex-

plore the suitability of the new modular structure presented in [57] induced from the novel module extraction technique based on datalog reasoning [58]. Third, we will investigate the possibility of incremental reasoning based on incrementally updateable decomposition of OWL ontologies [43].

Acknowledgements We thank the anonymous referees for their critical comments on a previous version of this paper, which encouraged us to significantly improve the paper. This work was supported by the National Key Research and Development Program of China (2016YFB1000603), the National Natural Science Foundation of China (NSFC) (Grant No. 61672377), and the Key Technology Research and Development Program of Tianjin (16YFZCGX00210). Xiaowang Zhang is supported by Tianjin Thousand Young Talents Program.

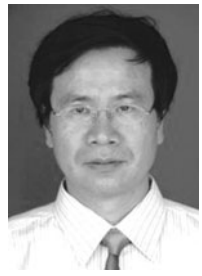
References

1. Horrocks I, Patel-Schneider P F, van Harmelen F. From SHIQ and RDF to OWL: the making of a web ontology language. Journal of Web Semantics, 2003, 1(1): 7–26
2. Patel-Schneider P, Hayes P, Horrocks I. Web ontology language OWL abstract ayntax and aemantics. W3C Recommendation, 2004
3. Cuenca Grau B, Horrocks I, Motik B, Parsia B, Patel-Schneider P F, Sattler U. OWL 2: the next step for OWL. Journal of Web Semantics, 2008, 6(4): 309–322
4. Motik B, Patel-Schneider P F, Cuenca Grau B. OWL 2 Web ontology language direct semantics. W3C Recommendation, 2009
5. Berners-Lee T, Hendler J, Lassila O. The semantic Web. Scientific American, 2001, 284(5): 28–37
6. Sidhu A, Dillon T, Chang E, Sidhu B S. Protein ontology development using OWL. In: Proceedings of the 1st Workshops on OWL: Experiences and Directions. 2005
7. Golbreich C, Zhang S, Bodenreider O. The foundational model of anatomy in OWL: experience and perspectives. Journal of Web Semantics, 2006, 4(3): 181–195
8. Rector A, Rogers J. Ontological and practical issues in using a description logic to represent medical concept systems: experience from GALEN. In: Proceedings of the 2nd International Summer School on Reasoning Web. 2006, 197–231
9. Soergel D, Lauser B, Liang A, Fisseha F, Keizer J, Katz S. Reengineering thesauri for new applications: the AGROVOC example. Journal of Digital Information, 2006, 4(4): 1–23
10. Derriere S, Richard A, Preite-Martinez A. An ontology of astronomical object types for the virtual observatory. In: Proceedings of the 26th meeting of the IAU on Virtual Observatory in Action: New Science, New Technology, and Next Generation Facilities. 2006
11. Lacy L, Aviles G, Fraser K, Gerber W, Mulvehill A, Gaskill R. Experiences using OWL in military applications. In: Proceedings of the 1st Workshop on OWL: Experiences and Directions. 2005
12. Goodwin J. Experiences of using OWL at the ordnance survey. In: Proceedings of the 1st Workshop on OWL: Experiences and Directions. 2005
13. Lécué F, Schumann A, Sbodio M L. Applying semantic web technologies for diagnosing road traffic congestions. In: Proceedings of the 11th International Semantic Web Conference. 2012, 114–130
14. Lécué F, Tucker R, Bicer V, Tommasi P, Tallevi-Diotallevi S, Sbodio M. Predicting severity of road traffic congestion using semantic Web technologies. In: Proceedings of the 11th Extended Semantic Web Conference. 2014, 611–627
15. Kazakov Y, Krötzsch M, Simancík F. Concurrent classification of *EL* ontologies. In: Proceedings of the 10th International Semantic Web Conference. 2011, 305–320
16. Glimm B, Horrocks I, Motik B, Shearer R, Stoilos G. A novel approach to ontology classification. Journal of Web Semantics, 2011, 14(1): 84–101
17. Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P. The description logic handbook: theory, implementation, and applications. Cambridge: Cambridge University Press, 2007
18. Kazakov Y. RIQ and SROIQ are harder than SHOIQ. In: Proceedings of the 11th International Conference on Knowledge Representation and Reasoning. 2008, 274–284
19. Horrocks I, Sattler U. A tableau decision procedure for SHOIQ. Journal of Automated Reasoning, 2007, 39(3): 249–276
20. Motik B, Shearer R, Horrocks I. Hypertableau reasoning for description logics. Journal of Artificial Intelligence Research, 2009, 36: 165–228
21. Glimm B, Horrocks I, Motik B, Stoilos G, Wang Z. HermiT: an OWL 2 reasoner. Journal of Automated Reasoning, 2014, 53(3): 245–269
22. Tsarkov D, Horrocks I. FaCT++ description logic reasoner: system description. In: Proceedings of the 3rd International Joint Conference on Automated Reasoning. 2006, 292–297
23. Haarslev V, Möller R. Racer System description. In: Proceedings of the 1st International Joint Conference on Automated Reasoning. 2001, 701–705
24. Sirin E, Parsia B, Cuenca Grau B, Kalyanpur A, Katz Y. Pellet: a practical OWL DL reasoner. Journal of Web Semantics, 2007, 5(2): 51–53
25. Goncalves R S, Parsia B, Sattler U. Performance heterogeneity and approximate reasoning in description logic ontologies. In: Proceedings of the 11th International Semantic Web Conference. 2012, 82–98
26. Krözsch M. OWL 2 profiles: an introduction to lightweight ontology languages. In: Proceedings of the 8th Reasoning Web Summer School. 2012, 112–183
27. Baader F, Brandt S, Lutz C. Pushing the *EL* envelope. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence. 2005, 364–369
28. Harris M A, Clark J, Ireland A. Gene ontology consortium: the gene ontology (GO) database and informatics resource. Nucleic Acids Research, 2004, 32: 258–261
29. Spackman K A. Rates of change in a large clinical terminology: three years experience with snomed clinical terms. In: Proceedings of the AMIA Annual Symposium. 2005, 714–718
30. Mendez J, Suntisrivaraporn B. Reintroducing CEL as an OWL 2 *EL* reasoner. In: Proceedings of the 22nd International Workshop on Description Logics. 2009
31. Mendez J. JCel: a modular rule-based reasoner. In: Proceedings of the 1st International Workshop on OWL Reasoner Evaluation. 2012, 858
32. Kazakov Y, Krözsch M, Simančik F. The incredible ELK. Journal of Automated Reasoning, 2014, 53(1): 1–61
33. Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, Goldberg L J, Eilbeck K, Ireland A, Mungall C J, OBI Consortium, Leontis N,

Rocca-Serra P, Ruttenberg A, Sansone S A, Scheuermann R H, Shah N, Whetzel L, Lewis S. The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology, 2007, 25(11): 1251–1255

34. Sioutos N, De Coronado S, Haber M W, Hartel F W, Shaiu W L, Wright L W. NCI thesaurus: a semantic model integrating cancer-related clinical and molecular information. Journal of Biomedical Informatics, 2007, 40: 30–43
35. Armas Romero A, Cuenca Grau B, Horrocks I. MORe: modular combination of OWL Reasoners for ontology classification. In: Proceedings of the 11th International Semantic Web Conference. 2012, 1–16
36. Tsarkov D, Palmisano I. Divide Et Impera: metareasoning for large ontologies. In: Proceedings of the 9th Internation Workshop on OWL: Experiences and Directions. 2012
37. Song W, Spencer B, Du W. Complete classification of complex ALCHO ontologies using a hybrid reasoning approach. In: Proceedings of the 26th International Workshop on Description Logics. 2013, 942–961
38. Steigmiller A, Glimm B, Liebig T. Coupling tableau algorithms for expressive description logics with completion-based saturation procedures. In: Proceedings of the 7th International Joint Conference on Automated Reasoning. 2014, 449–463
39. Angeli G, Nayak N, Manning G D. Combining natural logic and shallow reasoning for question answering. Technical Report in The Stanford Natural Language Processing Group, 2016
40. Del Vescovo C, Parsia B, Sattler U, Schneider T. The modular structure of an ontology: atomic decomposition. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence. 2011, 2232–2237
41. Del Vescovo C, Parsia B, Sattler U. Topicality in logic-based ontologies. In: Proceedings of the 19th International Conference on Conceptual Structures. 2011, 25–29
42. Del Vescovo C, Parsia B, Sattler U. Logical relevance in ontologies. In: Proceedings of the International Workshop on Description Logics. 2012
43. Klinov P, Del Vescovo C, Schneider T. Incrementally updateable and persistent decomposition of OWL ontologies. In: Proceedings of the 9th International Workshop on OWL: Experiences and Directions. 2012
44. Horridge M, Mortensen J M, Parsia B, Sattler U, Musen M A. A study on the atomic decomposition of ontologies. In: Proceedings of the 13th International Semantic Web Conference. 2014, 65–80
45. Wang C L, Feng Z Y. A novel combination of reasoners for ontology classification. In: Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence. 2013, 463–468
46. Cuenca Grau B, Horrocks I, Kazakov Y, Sattler U. Modular reuse of ontologies: theory and practice. Journal of Artificial Intelligence Researchvol, 2008, 31(1): 273–318
47. Cuenca Grau B, Halaschek-Wiener C, Kazakov Y, Suntisrivaraporn B. Incremental classification of description logics ontologies. Journal of Automated Reasoning, 2010, 44(4): 337–369
48. Del Vescovo C, Gessler D D, Klinov P, Parsia B, Sattler U, Schneider T, Winget A. Decomposition and modular structure of bioportal ontologies. In: Proceedings of the 10th International Semantic Web Conference. 2011, 130–145
49. Del Vescovo C. The Modular structure of an ontology: atomic decomposition and its applications. Dissertation for the Doctoral Degree. Manchester: The University of Manchester, 2013
50. Simancik F, Kazakov Y, Horrocks I. Consequence-based reasoning beyond horn ontologies. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence. 2011, 1093–1098
51. Martín-Recuerda F, Walther D. Fast modularisation and atomic decomposition of ontologies using axiom dependency hypergraphs. In: Proceedings of the 13th International Semantic Web Conference. 2014, 49–64
52. Groot P, Stuckenschmidt H, Wache H. Approximating description logic classification for semantic Web reasoning. In: Proceedings of the 2nd European Semantic Web Conference. 2005, 318–332
53. Kazakov Y. Consequence-driven reasoning for horn SHIQ ontologies. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence. 2009, 2040–2045
54. Lembo D, Santarelli V O, Fabio Savo D. A graph-based approach for classifying OWL 2 QL ontologies. In: Proceedings of the 26th International Workshop on Description Logics. 2013, 747–759
55. Liu Z H, Feng Z Y, Zhang X W, Wang X, Rao G Z. RORS: enhanced rule-based OWL reasoning on Spark. In: Proceedings of the 18th Asia-Pacific Web Conference on Web Technologies and Applications. 2016, 444–448
56. Liu Z H, Ge W, Zhang X W, Feng Z Y. Enhancing rule-based OWL reasoning on spark. In: Proceedings of the 15th International Semantic Web Conference (Posters & Demonstrations Track). 2016
57. Wang C L, Feng Z Y, Rao G Z, Wang X, Zhang X W. From datalog reasoning to modular structure of an ontology. In: Proceedings of the 14th International Semantic Web Conference (Posters & Demonstrations Track). 2015
58. Armas Romero A, Kaminski M, Cuenca Grau B, Horrocks I. Ontology module extraction via datalog reasoning. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence. 2015, 1410–1416

Changlong Wang received his MS degree from Northwest Normal University, China in 2010. He is currently a PhD candidate in School of Computer Science and Technology at Tianjin University, China. His current research interests are in semantic Web, knowledge representation and reasoning, and ontology.

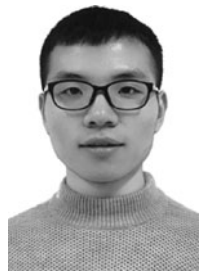


Zhiyong Feng received his PhD degree from Tianjin University (TJU), China in 1996. Since 1996, he has been with TJU, where he is currently a professor of the School of Computer Science and Technology. His current research interests are in knowledge engineering, service computation, and cognitive computation. He is a member of ACM and IEEE.

Xiaowang Zhang received his PhD degree from Peking University, China in 2011. Since 2015, he has been with Tianjin University, where he is currently an associate professor in School of Computer Science and Technology. His current research interests are in knowledge graph, graph databases, and knowledge representation and reasoning. He is a member of ACM and CCF.

Xin Wang received his PhD degree in computer science and technology from Nankai University, China in 2009. Since 2009, he has been with Tianjin University, where he is currently an associate professor in School of Computer Science and Technology. His research interests are in semantic data management, graph databases, and large-scale knowledge processing.

Guozheng Rao received his PhD degree from Tianjin University, China in 2009. Since 2000, he has been with Tianjin University, where he is currently an associate professor in School of Computer Science and Technology. His current research interests are in knowledge engineering, ontology, and semantic Web.



Daoxun Fu is currently a master student in the School of Computer Science and Technology at Tianjin University, China. His current research interests are in query answering, and knowledge representation and reasoning.