# Empirical evaluation of reasoning in lightweight DLs on life science ontologies

Boontawee Suntisrivaraporn
Theoretical Computer Science
University of Technology Dresden
Dresden, Germany
meng@tcs.inf.tu-dresden.de

***Abstract*—Description Logics (DLs) belong to a successful family of knowledge representation formalisms with two key assets: formally well-defined *semantics* which allows to represent knowledge in an unambiguous way and automated *reasoning* which allows to infer implicit knowledge from the one given explicitly. One of the most prominent applications of DLs is their use as ontology languages, especially for the life science domain. This paper investigates several life science ontologies and summarizes their common characteristics. It suggests that the use of lightweight DLs in the $\mathcal{EL}$ family, in which reasoning is tractable, is beneficial both in terms of expressivity and of scalability. The claim is supported by extensive empirical evaluation of various DL reasoning services on large-scale life science ontologies, including an overview comparison of state-of-the-art DL reasoners.**

***Index Terms*—Description Logics; Tractable reasoning; Life science ontologies;**

## I. Introduction

Description Logics (DLs) have evolved from the early knowledge representation formalisms of *semantic networks* and *frames*. Both predecessors of DLs share the notions of classes of individuals and relations between such classes.

These notions are realized in semantic networks as *vertices* and *edges* in a labeled directed graph. Vertices represent either *individuals* or *classes* of individuals (also called *concepts*), and labeled edges represent *relations* between them. A special type of relation, called is-a, is used in semantic networks to specify the generality or specificity of classes. Other kinds of relationships are realized as edges with other labels, e.g., has-color in Figure 1. In frame systems, concepts are realized as *frames* similar to the notion of classes in object-oriented programming languages. Each frame has a name, a collection of more general frames and a collection of *slots*. Slots are used to specify properties of concepts by linking the current frame to others in a similar sense as edges in semantic networks.

The main problem with both semantic networks and frame systems is that they lacked a formally well-defined semantics. For instance, it is unclear what the edge has-color in Figure 1 is intended to mean. One possible reading is that "frogs may only have color green," while another is that "frogs have at least a color green." Yet, the edge may be understood as a default property of frogs that can be overridden later when more knowledge is specified. Moreover, allowing vertices to represent both individuals and classes of individuals is ambiguous (e.g., Harry is intended to be an individual frog as opposed to Frog and Amphibian). Having to rely on their own *operational semantics*, different reasoning algorithms for the same formalism could behave differently upon the same knowledge base. To overcome this problem, *declarative semantics* had to be defined formally and independently of any specific reasoning algorithms. Attempts to employ fragments of first-order logics in these early KR systems have eventually resulted in 'logic-based concept languages' which have later become known as Description Logics.

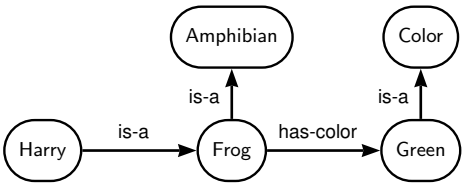


Fig. 1. An example of a semantic network.

### A. The quest for tractable Description Logics

The quest for tractable (i.e., polynomial-time decidable) Description Logics started in the 1980s after the first intractability results for DLs were shown [10]. Until relatively recently, it was restricted to DLs that extend the basic language $\mathcal{FL}_0$, which comprises the concept constructors conjunction and universal quantification. The main reason for this focussing was that, when clarifying the logical status of property edges in semantic networks and slots in frames, the decision was taken that edges and slots should be read as universal quantifications rather than existential quantifications. In our example, the edge has-color would read that "frogs may only have color green."

In most applications of DLs, it is crucial to reason with *terminologies* or *TBoxes*, rather than with isolated concept descriptions. Unfortunately, as soon as terminologies are taken into consideration, tractability turns out to be unattainable in $\mathcal{FL}_0$. Classifying even the simplest form of terminologies (known as *acyclic* or *unfoldable* TBoxes) that admit only acyclic concept definitions was shown to be coNP-hard [24]. If the most general form of terminologies is admitted (known as *general* TBoxes), which consists of general concept inclusion (GCI) axioms as supported by all modern DL systems, then classification in $\mathcal{FL}_0$ even becomes ExpTime-complete [2]. For these reasons, and also because of the need for expressive

1-4244-2384-2/08/$20.00  

DLs in applications, from the mid 1990s on, the DL community has mainly given up on the quest of finding tractable DLs. Instead, it investigated more and more expressive DLs, for which reasoning is worst-case intractable. The goal was then to find practical reasoning procedures, i.e., algorithms that are easy to implement and optimize, and which—though worst-case exponential or even worse—behave well in practice (see, e.g., [16]). This line of research has resulted in the availability of highly optimized DL systems for expressive Description Logics based on tableau algorithms [16], [14], and in successful applications—most notably is the recommendation by the W3C of the DL-based *Web Ontology Language* (better known as *OWL*) [17] as the ontology language for the Semantic Web.

At the beginning of the present decade, the choice of value restrictions as a *sine qua non* of DLs has been reconsidered. On the one hand, it was shown that the DL $\mathcal{EL}$, which allows for conjunction and existential restrictions, has better algorithmic properties than $\mathcal{FL}_0$. Precisely, classification of both acyclic and cyclic $\mathcal{EL}$ TBoxes is tractable [5], and this remains so even if general TBoxes with GCIs are admitted [11]. On the other hand, there are applications where value restrictions are not needed, and where the expressive power of $\mathcal{EL}$ or its small extensions appear to be sufficient. In fact, the Systematized Nomenclature of Medicine, Clinical Terms, [29] employs $\mathcal{EL}$ with an acyclic TBox extended with role inclusion axioms. Also, the Gene Ontology [1], the thesaurus of the US National Cancer Institute and the Foundational Model of Anatomy can be seen as acyclic $\mathcal{EL}$ TBoxes. Finally, large parts of the GALEN Medical Knowledge Base [27] can also be expressed in $\mathcal{EL}$ with GCIs, role hierarchy, and transitive roles.

### B. DL systems

Tableau-based algorithms for expressive DLs have been optimized and implemented in the DL reasoning systems FaCT [16] and Racer [14]. These implementations used several optimization techniques including the ones developed in [6], [16]. With the highly effective optimization techniques, these reasoning systems turned out to perform surprisingly well on TBoxes from practical applications. It has been observed that hard cases leading to the worst-case behaviors of the algorithms rarely occurred in practice (see, e.g, [16]). This observation encouraged research on pushing expressivity of DLs further and developing practical algorithms.

Current tableau-based DL systems, such as FaCT$^{++}$, RacerPro and Pellet, not only offer more expressive DLs (i.e., up to $\mathcal{SROIQ}$ [19] which is the logical underpinning of the new Web Ontology Language OWL 2) but also employ additional optimizations that have been tailored toward specific applications like biomedical ontologies (see, e.g., [18], [15]). Alternative DL systems include KAON2 [22], which implemented an algorithm based on resolution reasoning and disjunctive datalog; and HermiT [23], which implemented a novel calculus known as 'hypertableau.'

Distinguishingly, the CEL reasoner [4] supports the lightweight DL $\mathcal{EL}^+$, a useful, tractable extension of $\mathcal{EL}$. At first sight, one might think that a polynomial-time algorithm is always better suited for implementation than worst-case exponential-time algorithms such as the ones underlying those modern DL reasoners. However, due to the plethora of sophisticated optimization techniques that have been developed for tableau algorithms over the last decade, it is far from obvious whether a straightforward implementation of the polynomial-time algorithm can compete with highly-optimized implementations of tableau algorithms. A case in point is our experience with implementing the polynomial-time classification algorithms for cyclic $\mathcal{EL}$ TBoxes introduced in [5]: direct implementations of both the algorithm for subsumption w.r.t. descriptive semantics (based on a reduction to satisfiability of propositional Horn formulae) and the algorithm for subsumption w.r.t. greatest fixpoint semantics (based on computing the greatest simulation on a graph) did not lead to satisfactory results on the Gene Ontology [30].

CEL implemented a *refined* polynomial-time classification algorithm [7], where an obvious obstacle for efficient implementation of the algorithm given in [2] is removed—namely, the uninformed, brute-force search for applicable completion rules. With almost no further optimizations, the first implementation has demonstrated high performance on specific applications of biomedical ontologies [7]. Not only have these empirical results of CEL encouraged the use of the tractable DL family of $\mathcal{EL}$, but they have also sparked interest in further research into new optimization techniques for expressive DLs. By taking into account the underlying logic of the input TBox, a tableau-based reasoner can wisely select the most optimal algorithm and/or enable specific optimizations to perform reasoning (see, e.g., [18], [15]).

Besides the standard reasoning of classification, CEL supports incremental classification, module extraction and axiom pinpointing. The reasoning techniques implemented in the CEL system are described in detail in the author's PhD thesis [32].

### C. Formal ontologies

In the context of knowledge representation and reasoning, *(formal) ontologies* are specifications of conceptualization. Hence, a terminology in the DL sense, e.g., a general TBox, can be seen as a formal ontology. Ontologies based on Description Logics can be used, for example, to formalize technical vocabularies and to perform semantic indexing and query answering in variety of applications, including the Semantic Web and life science.

In this paper, ontologies from life science are considered due to their shared characteristics that match the DL under consideration. The next section investigates a few biomedical ontologies and discusses their common characteristics and the challenge of reasoning with them.

## II. Life Science Ontologies

Given the vast knowledge of biology and medicine acquired even before the advent of computing systems, not to mention the complexity of this knowledge, it is not astonishing that researchers in these scientific branches have encountered the problem of representing their knowledge in a systematic way.

Several efforts to systematize biomedical knowledge and standardize terms have eventually resulted in either classifications of diseases, controlled vocabularies, thesauri, terminologies or ontologies. By formalizing knowledge in an unequivocal way, the biomedical community can create a common understanding of the subject in the sense that it helps reduce redundancy in and heterogeneity of the domain knowledge.

The Systematized Nomenclature of Medicine, Clinical Terms (SNOMED CT) is a comprehensive clinical and medical ontology that covers a wide range of concepts in the domain, including anatomy, diseases, pharmaceutical products, clinical findings and medical procedures [29]. The presence of this terminology dated back to 1965 when the College of American Pathologist (CAP) released the Systematized Nomenclature of Pathology (SNOP) which was extended in 1997 to the first version of Systematized Nomenclature of Medicine, known as SNOMED Reference Terminology (RT) [28]. It was claimed to be the first version of this terminology to use the formal semantics (through the KRSS syntax [26]) of DLs. The terminology has since been continually revised and finally merged with Clinical Terms version 3 [25] to form the much more comprehensive terminology SNOMED CT, which comprises almost four hundred thousand concept definitions like

$$\begin{aligned}\mathsf{AmputationOfFinger} \equiv\ & \mathsf{HandExcision} \sqcap \\ & \exists \mathsf{roleGroup}.(\ \exists \mathsf{direct\text{-}procedure\text{-}site}.\mathsf{Finger}_S \sqcap \\ & \quad \exists \mathsf{method}.\mathsf{Amputation}\ )\end{aligned}$$

A small extension of the DL $\mathcal{EL}$ has so far been used as a primary language for development where automated reasoning of classification has proved useful in the generation of SNOMED CT in 'normal form' [28] for distribution purposes.

In 1992, the European project GALEN[1] was launched in order to facilitate the integration of medical information systems by means of a common reference model for medical terminology. Unlike the approach to SNOMED RT by the CAP, who translated an existing classification system for its previous version of the terminology into DL, the strategy of the GALEN project was to invent a suitable KR formalism before developing the actual terminology. Through compilation of specific requirements for the medical domain, the 'GALEN Representation and Integration Language (GRAIL)' was devised and used to develop the GALEN medical ontology [27].

In order to benchmark his DL reasoner FaCT, Horrocks [16] has translated the GALEN ontology into the DL format by proposing a mapping from GRAIL statements to equivalent logical axioms formulated in the DL $\mathcal{ALCH}f_{R^+}$ or $\mathcal{SH}f$. An investigation of GRAIL under scrutiny has revealed that it also supports so-called inverse roles, but this was not included in the ontology fragments used as benchmarks in [16]. The mapping can easily be extended to take into account inverse roles. Since concept disjunction, negation and universal quantification have not been included in the GRAIL language, the more fine-tuned DL for GALEN is $\mathcal{ELHI}f_{R^+}$ [35].

[1]Generalised Architecture for Languages, Encyclopaedias and Nomenclatures in Medicine; see http://www.OpenGALEN.org.

An interesting feature of GALEN that distinguishes it from most biomedical ontologies is that it makes use of GCIs which can be used to add levels of granularity and to supplement constraints. A classical example [20] for the former case is the use of a GCI like

$$\begin{aligned}\mathsf{Ulcer} \sqcap \exists \mathsf{has\text{-}loc}.\mathsf{Stomach} \quad & \sqsubseteq \\ \mathsf{Ulcer} \sqcap \exists \mathsf{has\text{-}loc}.(\mathsf{Lining} \sqcap & \exists \mathsf{part\text{-}of}.\mathsf{Stomach})\end{aligned}$$

to bridge the term 'ulcer of stomach' to the more fine-grained term 'ulcer of lining of stomach' since it is known that ulcer of the stomach is specific to the lining of the stomach.

Apart from SNOMED CT and GALEN, the repository of Open Biomedical Ontologies (OBO) is a large library of ontologies from the biological and medical domains. Most of the ontologies available in the repository have been written in 'OBO flat file format,' which was originally designed for the Gene Ontology (GO) [1]. The OBO format is relatively informal, but there have been attempts to map this format to Description Logic semantics. An example is given in [30] where two translations of the Gene Ontology were proposed, one of which turned out to correspond to the OBO's intended semantics.

More recently, Golbreich et al. has defined a semantic mapping from the OBO flat file format to OWL [12]. As a consequence of this mapping, several other biomedical ontologies readily available in the OBO format have been being translated into OWL. Similar to SNOMED CT and GALEN, biomedical ontologies developed using the OBO language turned out to be expressible in the DL $\mathcal{EL}$ or its tractable extensions. Unlike GALEN, however, they do not use GCIs and purely rely on concept definitions. Notable examples of OBO ontologies are the Gene Ontology, the thesaurus of the US National Cancer Institute (NCI) and the Foundational Model of Anatomy (FMA).

Common characteristics shared among biomedical ontologies can be summarized as follows:

- Disjunction, negation, universal quantification and cardinality restrictions are not explicitly required to formulate sensible ontologies in the biomedical domain. Only concept constructors in $\mathcal{EL}$, i.e., conjunction and existential quantification, appear to be adequate.
- Transitivity and role hierarchy axioms play an indispensable role in biomedical ontologies; while other role axioms, such as right-identity, functionality, domain and range restrictions, are sometimes required.
- As an inevitable consequence of the complexity of the domain, biomedical ontologies are typically of very large scale, comprising hundreds of thousands of concept definitions in some cases.

For most existing biomedical ontologies, the scalability of reasoning seems to outweigh the expressivity of the ontology language. In order to address these specific requirements, the DL $\mathcal{EL}^+$ and its reasoning techniques have been tailored toward ontologies of the kind [7], [32]—the language is sufficiently expressive for the applications at hand and reasoning can be accomplished in polynomial time.

## III. Tasks for Ontology Design and Maintenance

To identify the tasks for design and maintenance of formal ontologies, it is essential to cast some light on what makes good ontologies. The five basic principles for the design of formal ontologies proposed in [13] are summarized as follow:

- *Clarity*: an ontology should effectively convey the intended meaning of its defined terms. Full definitions providing necessary and sufficient conditions are preferred over primitive ones providing only necessary conditions.
- *Coherence*: an ontology should be logically consistent. Also, implicit consequences that contradict the domain knowledge should not be inferred from the ontology.
- *Extensibility*: an ontological structure should be so that it is possible to extend the ontology or refine some of its definitions monotonically.
- *Minimal encoding bias*: an ontology should be specified at the knowledge level, independent of specific encoding.
- *Minimal ontological commitment*: an ontology should make as few claims about the domain of discourse as possible, i.e., only terms essential for the intended use of the ontology are defined.

Note that DL-based knowledge representation systems promote good ontologies according to some of the above criteria. Clarity and minimal encoding criteria are attained as direct results from the well-defined formal semantics of Description Logics. Primitive and full concept definitions in DLs provide unambiguous utility to specify terms, while general concept inclusions allow to supplement additional constraints without having to interfere with existing definitions. Additionally, minimal encoding bias can be alleviated with the help of advanced ontology editors and visualization tools (e.g., Protégé and Swoop) that avoid hassles of a specific syntax (e.g., OWL) and thus help to promote coding at the knowledge level.

This paper investigates the roles of reasoning support in shaping good ontologies in terms of coherence and extendibility. In [21], the authors described tasks relevant for ontology design and maintenance, and argued how logical reasoning support can be used to accomplish them. Here, three tasks for ontology design and maintenance are considered:

*1) Authoring concept definitions:* One of the most central activities during ontology design and maintenance is the formulation of new concept definitions (design) and the refinement of existing concept definitions (maintenance). Due to the *declarative* style of DL semantics, the ontology developer cannot use some execution model to guide his intuition about the effects of design decisions. Unwanted implicit consequences may be incurred without awareness of the developer and can be far from easy to detect by hand.

Such implicit consequences could be that the ontology is logically inconsistent (i.e., there is no model); that a concept in the ontology is unsatisfiable (i.e., it cannot be instantiated); or that one concept is a subconcept of another. The first two types of consequences immediately indicate flaws in the ontology since an ontology is intended to represent at least a possible model, and a concept to represent a class of objects. Subsumptions may or may not be intended depending on its intuition in the domain of discourse. At any rate, they need to be detected and reported to the domain expert for inspection.

The mentioned tasks directly correspond to the reasoning problems of *consistency*, *satisfiability* and *subsumption* in DL. Most DL systems usually support *classification* which is the computation of the *subsumption hierarchy*. Not only is classification useful in detecting unwanted subsumptions, it also provides with a visualization of the ontology's structure and is the premier way to navigate and access the ontology. Classification is normally implemented by means of multiple subsumption checks, therefore it is of utmost importance for ontology maintenance that such a computation can be done incrementally when a small change is applied, i.e., previous classification information is reused.

*2) Error management:* Similar to writing large software programs, building large-scale ontologies is an error-prone endeavor. The aforementioned reasoning support can help alert the developer to the existence of errors. For example, 'amputation of finger' is inferred to be a subconcept of 'amputation of hand' in SNOMED CT, which is clearly unintended [33] and reveals a modeling error. However, given an unintended subsumption relationship in a large ontology like SNOMED CT with almost four hundred thousand axioms, it is not always easy to find the erroneous axioms responsible for it by hand.

Automated reasoning support for error management comes in three flavors: pinpointing, explanation and revision. *Pinpointing* identifies those concept definitions responsible for an error, while *explanation* aims to provide a convincing argument that also involves explaining the interplay between the relevant concept definitions. *Automatic revision* goes one step further by making concrete suggestions for how to resolve the error.

*3) Ontology import:* One of the first decisions to be made when building an ontology is whether to start from scratch or to reuse available knowledge in existing ontologies. For example, when building an ontology describing medicinal products of a pharmaceutical company, concepts of specific medical substances and human body parts may be used. In order to guarantee certain relationships among those concepts, the designer may want to include more details about them. Since these details have already been formulated properly in a standardized ontology like SNOMED CT, it should be less time consuming and more accurate to *import* the ontology. The problem, however, is that such standardized ontologies are typically designed to be comprehensive, thus very large, therefore importing the whole ontology unnecessarily introduces overhead in computation.

It is thus helpful to be able to extract a small portion of the ontology that contains only concept definitions relevant to the needs, i.e., knowledge about the concepts to be imported. To this end, the automated reasoning support of module extraction computes a subset of the ontology that is ensured to be small and adequately capture the meaning of the imported concepts.

| Name | Syntax | Semantics |
|---|---|---|
| top concept | $\top$ | $\Delta^{\mathcal{I}}$ |
| bottom concept | $\bot$ | $\emptyset$ |
| conjunction | $C \sqcap D$ | $C^{\mathcal{I}} \cap D^{\mathcal{I}}$ |
| existential quantification | $\exists r.C$ | $\{d \in \Delta^{\mathcal{I}} \mid \exists e\colon (d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$ |
| concept inclusion | $C \sqsubseteq D$ | $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ |
| role inclusion | $r_1 \circ \cdots \circ r_k \sqsubseteq s$ | $r_1^{\mathcal{I}} \circ \cdots \circ r_k^{\mathcal{I}} \subseteq s^{\mathcal{I}}$ |
| range restriction | $\mathsf{range}(r) \sqsubseteq C$ | $\{e \in \Delta^{\mathcal{I}} \mid \exists d\colon (d,e) \in r^{\mathcal{I}}\} \subseteq C^{\mathcal{I}}$ |
| concept assertion | $C(a)$ | $a^{\mathcal{I}} \in C^{\mathcal{I}}$ |
| role assertion | $r(a,b)$ | $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ |

TABLE I
SYNTAX AND SEMANTICS OF $\mathcal{EL}^+$.

## IV. THE DESCRIPTION LOGIC $\mathcal{EL}^+$

In DLs, *concept descriptions* are inductively defined with the help of a set of *constructors*, starting with a set $\mathsf{CN}$ of *concept names* and a set $\mathsf{RN}$ of *role names*. $\mathcal{EL}^+$ concept descriptions are formed using the constructors shown in the upper part of Table I.[2] An *unrestricted* $\mathcal{EL}^+$ *TBox* is a finite set of *general concept inclusions (GCIs)*, *role inclusions (RIs)* and *range restrictions*. Let $\mathsf{Ind}$ be a set of *individuals* disjoint from $\mathsf{CN}$ and $\mathsf{RN}$. Then, an $\mathcal{EL}^+$ *ABox* is a finite set of *concept assertions* and *role assertions*. Possibly with sub- or superscripts, $a, b, \ldots$ are conventionally used to range over individuals from $\mathsf{Ind}$; $r, s, \ldots$ over role names from $\mathsf{RN}$; $A, B$ over concept names from $\mathsf{CN}$; and $C, D$ over concept descriptions. The syntax elements of TBox axioms and ABox assertions are shown in the middle and lower part of Table I, respectively. An $\mathcal{EL}^+$ ontology $\mathcal{O}$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ such that the following syntactic restriction[3] is satisfied: if $r_1 \circ \cdots \circ r_k \sqsubseteq s \in \mathcal{O}$ with $k \geq 1$ and $\mathcal{O} \models \mathsf{range}(s) \sqsubseteq C$, then $\mathcal{O} \models \mathsf{range}(r_k) \sqsubseteq C$, where $\mathcal{O} \models \mathsf{range}(u) \sqsubseteq C$ if there is a role name $v$ such that $u \sqsubseteq_{\mathcal{O}} v$ and $\mathsf{range}(v) \sqsubseteq C \in \mathcal{O}$. If the assertional component is irrelevant, we sometimes use the terms terminology (TBox) and ontology interchangeably. The notations $\mathsf{CN}(\mathcal{O}), \mathsf{RN}(\mathcal{O}), \mathsf{Ind}(\mathcal{O})$ denote, respectively, the sets of concept names, role names and individuals occurring in $\mathcal{O}$, while $\mathsf{Sig}(\mathcal{O})$ denotes the signature of $\mathcal{O}$, i.e., $\mathsf{CN}(\mathcal{O}) \cup \mathsf{RN}(\mathcal{O}) \cup \mathsf{Ind}(\mathcal{O})$. Similarly, the notations are sometimes used with a concept description $C$ and an axiom/assertion $\alpha$. Figure 2 depicts an example $\mathcal{EL}^+$ ontology motivated by SNOMED CT and GALEN. Some remarks concerning the expressivity of $\mathcal{EL}^+$ are in order:

- A *primitive concept definition* $A \sqsubseteq C$, which specifies the necessary condition, is a special form of GCI, while a *concept definition* $A \equiv C$, which specifies the necessary and sufficient conditions, can be expressed by two GCIs: $A \sqsubseteq C$ and $C \sqsubseteq A$.

[2]Refer, e.g., to [3], [19] for other concept constructors.
[3]To ensure decidability and tractability of reasoning.

| | | | |
|---|---|---|---|
| $\alpha_1$ | Appendix | $\sqsubseteq$ | BodyPart $\sqcap$ $\exists$part-of.Intestine |
| $\alpha_2$ | Endocardium | $\sqsubseteq$ | Tissue $\sqcap$ $\exists$part-of.HeartValve $\sqcap$ $\exists$part-of.HeartWall |
| $\alpha_3$ | HeartValve | $\sqsubseteq$ | BodyValve $\sqcap$ $\exists$part-of.Heart |
| $\alpha_4$ | HeartWall | $\sqsubseteq$ | BodyWall $\sqcap$ $\exists$part-of.Heart |
| $\alpha_5$ | Appendicitis | $\equiv$ | Inflammation $\sqcap$ $\exists$has-loc.Appendix |
| $\alpha_6$ | Endocarditis | $\equiv$ | Inflammation $\sqcap$ $\exists$has-loc.Endocardium |
| $\alpha_7$ | Pancarditis | $\equiv$ | Inflammation $\sqcap$ $\exists$has-exact-loc.Heart |
| $\alpha_8$ | Inflammation | $\sqsubseteq$ | Disease $\sqcap$ $\exists$acts-on.Tissue |
| $\alpha_9$ | HeartDisease | $\equiv$ | Disease $\sqcap$ $\exists$has-loc.Heart |
| $\alpha_{10}$ | Tissue $\sqcap$ Disease | $\sqsubseteq$ | $\bot$ |
| $\alpha_{11}$ | HeartDisease $\sqcap$ $\exists$agent.Virus | $\sqsubseteq$ | ViralDisease $\sqcap$ $\exists$has-state.NeedsTreatment |
| $\alpha_{12}$ | has-exact-loc | $\sqsubseteq$ | has-loc |
| $\alpha_{13}$ | $\epsilon$ | $\sqsubseteq$ | part-of |
| $\alpha_{14}$ | part-of $\circ$ part-of | $\sqsubseteq$ | part-of |
| $\alpha_{15}$ | has-loc $\circ$ part-of | $\sqsubseteq$ | has-loc |

Fig. 2. An example $\mathcal{EL}^+$ ontology $\mathcal{O}_{\mathsf{med}}$.

- GCIs together with the bottom concept can be used to express *concept disjointness* as $C \sqcap D \sqsubseteq \bot$;
- A *domain restriction* is expressible as $\exists r.\top \sqsubseteq C$;
- Role inclusions generalize at least four important kinds of role axioms: (i) *role hierarchy axiom* $r \sqsubseteq s$, (ii) *reflexivity axioms* $\epsilon \sqsubseteq r$ with $\epsilon$ the nullary role composition, (iii) *transitivity axiom* $r \circ r \sqsubseteq r$ and (iv) *right identity axiom* $r \circ s \sqsubseteq r$. Examples of these role axioms are $\alpha_{12}$–$\alpha_{15}$, respectively.

The semantics of $\mathcal{EL}^+$ is defined in terms of *interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where the domain $\Delta^{\mathcal{I}}$ is a non-empty set of individuals, and the interpretation function $\cdot^{\mathcal{I}}$ maps each ABox individual $a \in \mathsf{Ind}$ to $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each concept name $A \in \mathsf{CN}$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, and each role name $r \in \mathsf{RN}$ to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$. The extension of $\cdot^{\mathcal{I}}$ to arbitrary concept descriptions is inductively defined as shown in the semantics column of Table I. An interpretation $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$ if, for each axiom and assertion in $\mathcal{O}$, the conditions given in the semantics column of Table I are satisfied.

The remainder of this section formally introduces reasoning services (a.k.a. inference problems) that correspond to or help to alleviate the tasks for ontology design and maintenance identified in the previous section. The reasoning support considered in this paper can be categorized into *standard* and *supplemental* reasoning services.

### A. Standard reasoning services

Let $\mathcal{O}$ be an ontology. Then, $\mathcal{O}$ is *consistent* if it has a model. A concept $C$ is *satisfiable w.r.t.* $\mathcal{O}$ if there is a model $\mathcal{I}$ of $\mathcal{O}$ such that $C^{\mathcal{I}}$ is not empty. Two concepts $C$ and $D$ are *disjoint w.r.t.* $\mathcal{O}$ if their conjunction $C \sqcap D$ is unsatisfiable w.r.t. $\mathcal{O}$, i.e., $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ in every model $\mathcal{I}$ of $\mathcal{O}$. In the example, all concept names are satisfiable w.r.t. $\mathcal{O}_{\mathsf{med}}$, but $\mathsf{Endocarditis} \sqcap \mathsf{Endocardium}$ is not.

The concept $C$ is *subsumed* by $D$ w.r.t. $\mathcal{O}$ (written, $\mathcal{O} \models C \sqsubseteq D$ or $C \sqsubseteq_{\mathcal{O}} D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{O}$. In this case, $C$ is said to be a *subsumee* or *subconcept*, while $D$ is said to be a *subsumer* or *superconcept*. The concepts $C$ and $D$ are *equivalent* w.r.t. $\mathcal{O}$ (written, $\mathcal{O} \models C \equiv D$ or $C \equiv_{\mathcal{O}} D$) if they subsume each other. Subsumption between concept descriptions can always be reduced to that between concept names using either concept definitions or GCIs. Precisely, $C \sqsubseteq_{\mathcal{O}} D$ if, and only if, $A \sqsubseteq_{\mathcal{O} \cup \{A \equiv C, B \equiv D\}} B$ if, and only if, $A \sqsubseteq_{\mathcal{O} \cup \{A \sqsubseteq C, D \sqsubseteq B\}} B$, where $A, B$ are fresh concept names not occurring in $\mathcal{O}$. In the example, it is not difficult to see that HeartDisease subsumes Endocarditis and Pancarditis w.r.t. $\mathcal{O}_{\mathsf{med}}$.

Since subsumption and (un)satisfiability are inter-reducible in $\mathcal{EL}^+$ [4] and unsatisfiable concepts seldom occur in biomedical ontologies, it is sensible to focus attention on subsumption.

*Classification* of $\mathcal{O}$ is the computation of subsumption relationships between all pairs of concept names in $\mathcal{O}$. The classification results are often represented in the so-called *subsumption hierarchy* which is essentially a *directed acyclic graph (DAG)* where vertices are concept names (strictly speaking, equivalence classes of concept names) and edges represent immediate subsumption relationships.

For other standard reasoning services involving individuals, such as *instance checking* and *realization*, refer to [3], [32].

These reasoning services are supported by most state-of-the-art DL systems. Some DL systems, including CEL, also support supplemental reasoning services useful in ontology design and maintenance.

### B. Supplemental reasoning services

As mentioned earlier, axiom pinpointing aims to compute the relevant axioms responsible for an erroneous consequence. Here, consequences of interest are unintended subsumptions (Note that unsatisfiability can be treated in an analogous way since it can be reduced to subsumption). Suppose that $\mathcal{O}$ is an ontology and $\sigma$ is a subsumption such that $\mathcal{O} \models \sigma$. Then, a subset $\mathcal{S} \subseteq \mathcal{O}$ is a *minimal axiom set (MinA) for $\sigma$ w.r.t. $\mathcal{O}$* if (i) $\mathcal{S} \models \sigma$ and (ii) for every $\mathcal{S}' \subset \mathcal{S}$, $\mathcal{S}' \not\models \sigma$. Note that MinAs need not be unique nor minimum relative to set cardinality. An example showing that there may be exponentially many MinAs for a subsumption is given in [8]. Considering the subsumption $\sigma = (\mathsf{Endocarditis} \sqsubseteq \mathsf{HeartDisease})$, it is not hard to verify that the sets $\mathcal{S}_1 = \{\alpha_2, \alpha_3, \alpha_6, \alpha_8, \alpha_9, \alpha_{14}\}$ and $\mathcal{S}_2 = \{\alpha_2, \alpha_4, \alpha_6, \alpha_8, \alpha_9, \alpha_{14}\}$ are all the MinAs for $\sigma$ w.r.t. $\mathcal{O}_{\mathsf{med}}$. Dually, a *maximal non-axiom set (MaNA) for $\sigma$ w.r.t. $\mathcal{O}$* is a subset $\mathcal{S} \subseteq \mathcal{O}$ such that (i) $\mathcal{S} \not\models \sigma$, and (ii) for every $\mathcal{S}' \supset \mathcal{S}$, $\mathcal{S}' \models \sigma$. Intuitively, a MaNA is a candidate for the new, revised ontology with the unwanted subsumption removed. The set complement of a MaNA corresponds to a so-called *diagnosis*, i.e., a minimal set of axioms needed to be removed in order to suppress the problematic subsumption. Since only a small number of axioms typically need to be removed, it is more convenient to compute diagnoses than MaNAs.

In what follows, let $\mathbf{S}$ be a signature (i.e., set of concept and role names), and $\sigma$ a potential consequence that may or may not hold in $\mathcal{O}$. Then, a subset $\mathcal{O}' \subseteq \mathcal{O}$ is a *module for $\sigma$ in $\mathcal{O}$* (for short, a *$\sigma$-module in $\mathcal{O}$*) whenever: $\mathcal{O} \models \sigma$ if, and only if, $\mathcal{O}' \models \sigma$. A subset $\mathcal{O}' \subseteq \mathcal{O}$ is a *module for a signature $\mathbf{S}$ in $\mathcal{O}$* (for short, an *$\mathbf{S}$-module in $\mathcal{O}$*) if, for every potential consequence $\sigma$ with $\mathsf{Sig}(\sigma) \subseteq \mathbf{S}$, $\mathcal{O}'$ is a $\sigma$-module in $\mathcal{O}$. Intuitively, a module in an ontology $\mathcal{O}$ is a subset $\mathcal{O}' \subseteq \mathcal{O}$ that preserves a potential consequence of interest or the potential consequences over a signature of interest. Several techniques have been proposed in the literature in order to *syntactically* extract modules from an ontology. The next section reports on the empirical evaluation of the so-called *reachability-based* modules [31].

## V. Empirical Evaluation

Techniques for both standard and supplemental reasoning services defined previously are not the main focus of the present paper, and interested readers are encouraged to refer to [7], [8], [31], [9] and to [32]. For the purpose of this paper, it suffices to mention that these techniques have been implemented in the CEL reasoner which was used in the empirical evaluation of reasoning in $\mathcal{EL}^+$ on large-scale biomedical ontologies. The first subsection describes some characteristics of the tested biomedical ontologies. Then, the testing methodologies, as well as empirical results, are presented and discussed in the last subsection.

### A. Ontology test suite

The biomedical ontologies introduced in Section II were used as benchmarks in the empirical evaluation of classification and various other reasoning services supported by the CEL reasoner.

*1) The Systematized Nomenclature of Medicine, Clinical Terms (*SNOMED CT*):* is a comprehensive medical and clinical ontology. It is an $\mathcal{EL}$ TBox augmented with 11 role hierarchy axioms and two (complex) role inclusions. This ontology as of release *January/2005*, denoted by $\mathcal{O}^{\text{SNOMED}}$, consists of 379 691 concept and 62 role names.

*2) The Galen Medical Knowledge Base (*GALEN*):* has been developed within an EU project that sought to produce a reference ontology in a specialized DL. The full version of this ontology contains 23 136 concept and 950 role names.[5] It is precisely based on $\mathcal{ELHI}f_{R^+}$. The DL $\mathcal{EL}^+$ however can express most of its axioms, namely 95.75%, and this fragment was obtained (henceforth, $\mathcal{O}^{\text{FULLGALEN}}$) for experimental purposes by dropping role inverse and functionality axioms. The resulting ontology can still be considered realistic and identical to the original one apart from a number of missing subsumption relationships involving the removed role axioms.

Since the full version is both large and complex to be handled by DL reasoners, a simplified version of GALEN has often been considered as a decent benchmark ontology

[4] ($\Leftarrow$) $C$ is unsatisfiable w.r.t. $\mathcal{O}$ if, and only if, $C \sqsubseteq_{\mathcal{O}} \bot$; ($\Rightarrow$) $C \sqsubseteq_{\mathcal{O}} D$ if, and only if, $A$ is unsatisfiable w.r.t. $\mathcal{O}$ extended with GCIs $A \sqsubseteq C$ and $A \sqcap D \sqsubseteq \bot$ with $A$ a fresh concept name.

[5] http://www.co-ode.org/galen

for testing DL reasoners. This version[6] has originally been produced by Horrocks to evaluate FaCT and KRIS in his PhD thesis [16]. It consists of 2 748 concept and 413 role names. Again, its $\mathcal{EL}^+$ fragment was considered, denoted by $\mathcal{O}^{\text{NotGalen}}$, by dropping role inverse and functionality axioms.

*3) The Gene Ontology (*GO*):* project is a collaborative effort to address the need for consistent descriptions of gene products in different databases. It has developed and is maintaining three controlled vocabularies (i.e., ontologies) that describe gene products in terms of their associated *biological processes*, *cellular components* and *molecular functions* in a species-independent manner. The ontology, denoted by $\mathcal{O}^{\text{Go}}$, is formulated as an $\mathcal{EL}$ TBox with a single transitive role part-of. The release of $\mathcal{O}^{\text{Go}}$ used in the experiments consists of 20 465 concept names.

*4) The Thesaurus of the US National Cancer Institute (*NCI*):* is a large and carefully designed ontology that has become a reference terminology covering areas of basic and clinical science. The knowledge represented in NCI includes the domains of diseases, drugs, anatomy, genes, gene products, techniques and biological processes, all with a cancer-centric focus in its content. It was originally designed to support coding activities across the National Cancer Institute and to facilitate translational research in cancer. In the experiments, we considered the 'OWLized' version[7] that is formulated precisely as an $\mathcal{EL}$ TBox augmented with domain and range restrictions. It primitively defines 27 652 concept names and refers to 70 role names, each of which is constrained by a pair of domain and range restrictions.

*5) The Foundational Model of Anatomy (*FMA*):* is an evolving ontology concerned with the formal representation of human anatomy. Its ontological framework can be applied and extended to other species.[8] FMA has four interrelated components: the anatomy taxonomy, the anatomical structural abstraction, the anatomical transformation abstraction and the metaknowledge ontology. The ontology, denoted by $\mathcal{O}^{\text{Fma}}$, is indeed a large $\mathcal{EL}$ TBox extended with two transitivity axioms, one for part-of and the other for has-part. The number of concept names is 75 139.

Table II summarizes the size and other pertinent characteristics of all the test-suite ontologies. Numbers of axioms are broken down into the following kinds: primitive concept definitions (PCDef), full concept definitions (CDef), general concept inclusions (GCI), role inclusion axioms (RI). The latter also includes domain and range restrictions if present.

| Ontologies | ♯Concepts | ♯Roles | ♯Axioms | | | |
|---|---|---|---|---|---|---|
| | | | PCDef | CDef | GCI | RI |
| $\mathcal{O}^{\text{Go}}$ | 20 465 | 1 | 19 465 | 0 | 0 | 1 |
| $\mathcal{O}^{\text{Nci}}$ | 27 652 | 70 | 27 635 | 0 | 0 | 140 |
| $\mathcal{O}^{\text{Fma}}$ | 75 139 | 2 | 75 139 | 0 | 0 | 2 |
| $\mathcal{O}^{\text{NotGalen}}$ | 2 748 | 413 | 2 030 | 695 | 408 | 442 |
| $\mathcal{O}^{\text{FullGalen}}$ | 23 136 | 950 | 13 149 | 9 968 | 1 951 | 1 016 |
| $\mathcal{O}^{\text{Snomed}}$ | 379 691 | 62 | 340 972 | 38 719 | 0 | 13 |

TABLE II
THE TEST SUITE OF REALISTIC BIOMEDICAL ONTOLOGIES.

### B. Testing methodology and empirical results

The current version of CEL[9] is written in Common Lisp and compiled and built using Allegro Common Lisp 8.1. Like most evaluation methods for DL and other reasoning systems, all the experiments described in this section use 'CPU time' as the main performance indicator. Memory consumption is also discussed whenever appropriate. In order to confine the execution environment and hence to induce sensible comparison, the experiments were performed on the same Linux testing server which was equipped with a couple of 2.19GHz AMD Opteron processors and 2 GB of physical memory.

In the following, the testing methodology and the empirical results of each of the following reasoning services are described and discussed:

- classification,
- incremental classification,
- subsumption query answering,
- modularization, and
- axiom pinpointing.

*1) Classification:* Since classification is one of the most classical inference services, it is supported by all modern DL systems. For this reason, classification time is often used as a performance indicator for DL systems. A number of state-of-the-art DL reasoners—i.e., FaCT$^{++}$[10], HermiT[11], KAON2[12], Pellet[13] and RacerPro[14]—were considered for performance comparison. These DL reasoners vary in the sense that they implement different reasoning calculi and are written in different languages. For HermiT, KAON2 and Pellet, Sun's Java Runtime Environment (JRE) version 1.6.0 was used with alloted 1.5GB heap space. Some reasoners are not equipped with a profiling facility to internally measure CPU time. To achieve comparable measurement, an external timing utility was used with all the classifying systems.

All ontologies in the test suite described in the previous section were used as benchmarks for comparing the performance of the DL reasoners. Since KAON2's parser, and hence HermiT's parser, does not support (an extension of) the KRSS syntax [26], ontologies in the OWL format were used in their experiments. In the case of $\mathcal{O}^{\text{Snomed}}$, the two complex role inclusions were only passed to CEL and FaCT$^{++}$ but not to the other reasoners, as the latter do not support such axioms. Additionally, we needed to rename all the roles, because SNOMED uses the same codes for both roles and 'attributive' concepts but KAON2 and HermiT do not support such name punning. It has to be noted however that such renaming could

[6] http://www.cs.man.ac.uk/~horrocks/OWL/Ontologies/galen.owl
[7] http://www.mindswap.iorg/2003/CancerOntology
[8] http://sig.biostr.washington.edu/projects/fm/AboutFM.html
[9] http://lat.inf.tu-dresden.de/systems/cel/
[10] http://code.google.com/p/factplusplus/
[11] http://web.comlab.ox.ac.uk/people/Boris.Motik/HermiT/
[12] http://kaon2.semanticweb.org/
[13] http://pellet.owldl.com/
[14] http://www.racer-systems.com/

| Ontologies | $\mathcal{O}^{\text{GO}}$ | $\mathcal{O}^{\text{NCI}}$ | $\mathcal{O}^{\text{FMA}}$ | $\mathcal{O}^{\text{NOTGALEN}}$ | $\mathcal{O}^{\text{FULLGALEN}}$ | $\mathcal{O}^{\text{SNOMED}}$ |
|---|---|---|---|---|---|---|
| CEL | 0.98 | 3.75 | 9.04 | 2.83 | 201 | 1 258 |
| FaCT$^{++}$ | 20.12 | 1.72 | *t/o* | 3.28 | *m/o* | 606 |
| HermiT | 16.75 | 34.92 | 123 | 12.35 | *m/o* | *m/o* |
| KAON2 | *m/o* | *m/o* | *t/o* | *m/o* | *m/o* | *t/o* |
| Pellet | 52.58 | 36.11 | 7 753 | 31.56 | *m/o* | *m/o* |
| RacerPro | 17.11 | 13.36 | 629 | 17.06 | *t/o* | 1 155 |

TABLE III
COMPUTATION TIME (SECOND).

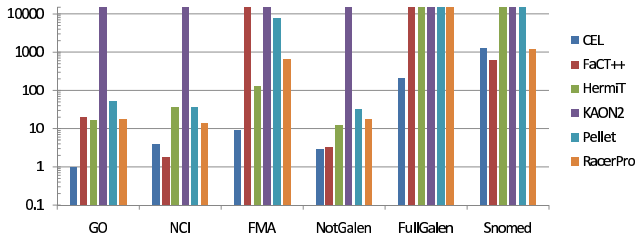


Fig. 3. Performance comparison through classification time (second).

by no means affect the meaning nor the classification results of the ontology. Table III shows the (two-run average) time taken by the respective reasoners to classify the biomedical ontologies, where *m/o* means that the reasoner failed due to memory exhaustion, and *t/o* means that the reasoner did not terminate within the allocated time of 24 hours. Figure 3 depicts a comparison chart of reasoners' performance based on their classification time, where both *m/o* and *t/o* are displayed as full vertical bars.

It can be seen from the chart and the table that CEL is the only DL reasoner that can classify all six biomedical ontologies in the test suite and outperforms HermiT, KAON2 and Pellet in all cases. Compared with the other reasoners, CEL is faster than RacerPro w.r.t. all but $\mathcal{O}^{\text{SNOMED}}$, and faster than FaCT$^{++}$ w.r.t. all but $\mathcal{O}^{\text{NCI}}$ and $\mathcal{O}^{\text{SNOMED}}$. It should be noted that, when it first came into existence in 2005 [7], CEL was the only academic DL system that was capable of classifying entire SNOMED CT. This has subsequently sparked interest in the DL community to research on optimization techniques specific to the biomedical ontologies (in particular, to SNOMED CT), and later enabled tableau-based reasoners like FaCT$^{++}$ and RacerPro to take advantage of simple structures of ontologies of this kind. These reasoners employed some of the optimization techniques described in [18], [15] that are highly effective on simpler TBoxes (i.e., without GCIs) like $\mathcal{O}^{\text{SNOMED}}$. When a large number of GCIs are present as in the case of $\mathcal{O}^{\text{FULLGALEN}}$, however, these reasoners fail due to either memory exhaustion or time out. Interestingly, CEL is the only reasoner that can classify $\mathcal{O}^{\text{FULLGALEN}}$.

HermiT and Pellet can classify the first four ontologies but fail on the last two, both due to a memory problem. The HermiT reasoner, which implements the much less non-deterministic hypertableau calculus [23], shows a relatively good performance. In fact, it noticeably outperforms Pellet in all cases and is even faster than FaCT$^{++}$ and RacerPro

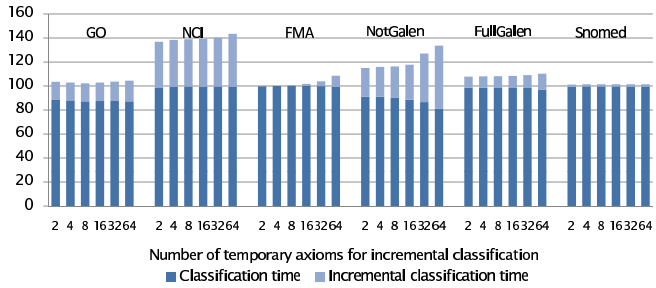


Fig. 4. Relative (incremental) classification time w.r.t. full classification time (percentage).

on some ontologies. KAON2 cannot classify any ontologies of this scale, but it is fair to remark that this DL system has been designed to deal with and optimized for conjunctive queries w.r.t. a large number of individuals.

In what follows, the testing methodology and empirical results for incremental classification, subsumption query answering, modularization and axiom pinpointing are described. In these experiments, only the CEL system was considered.

*2) Incremental classification:* To simulate usage scenarios of incremental classification, each ontology $\mathcal{O}$ in the test suite was partitioned into a permanent ontology and a temporary one. Ten repetitions of the following operations for each tested ontology $\mathcal{O}$ and each number $n = 2, 4, 8, 16, 32, 64$ were performed: *(i)* partition $\mathcal{O}$ into $\mathcal{O}_p$ and $\mathcal{O}_t$ such that the latter consist of $n$ random concept axioms from $\mathcal{O}$; *(ii)* classify $\mathcal{O}_p$ normally; and finally, *(iii)* incrementally classify $\mathcal{O}_t$ against $\mathcal{O}_p$. The time required to compute steps (ii) and (iii) was measured. The 20% trimmed average classification and incremental classification times of the 10 repetitions are considered, and their percentage relative to the full classification time of the whole ontology $\mathcal{O}$ is visualized in Figure 4. For each bar on the chart, the dark blue slice represents the *relative* classification time of $\mathcal{O}_p$, while the light blue slice represents the *relative* incremental classification time of $\mathcal{O}_t$ against $\mathcal{O}_p$. Hence, the entire bar depicts the overall computation time.

Classification of $\mathcal{O}_p$ took less time than the entire ontology $\mathcal{O}$, since the former is a subset of the latter. The time required to incrementally classify $\mathcal{O}_t$ varied, depending on the ontology and according to the number of new axioms. As a rule, the larger $\mathcal{O}_t$ is, the more time it took to incrementally classify it and also the more time it took overall to classify $\mathcal{O}_p$ and incrementally classify $\mathcal{O}_t$. The overall computation time (i.e., the height of each bar) was less than 150% for all ontologies and all numbers $n$. In the case of $\mathcal{O}^{\text{GO}}$, $\mathcal{O}^{\text{FMA}}$, $\mathcal{O}^{\text{FULLGALEN}}$ and $\mathcal{O}^{\text{SNOMED}}$, at most only 10% additional time was needed in order to incrementally classify up to 64 additional axioms. The proportion of incremental classification time is larger for $\mathcal{O}^{\text{NOTGALEN}}$ than other ontologies since the ontology itself is much smaller. In fact, 64 axioms already constitute more than 2% of the entire ontology. Though the size of $\mathcal{O}^{\text{NCI}}$ is in the same range as that of $\mathcal{O}^{\text{GO}}$ and $\mathcal{O}^{\text{FULLGALEN}}$, its relative incremental classification time was much greater. This

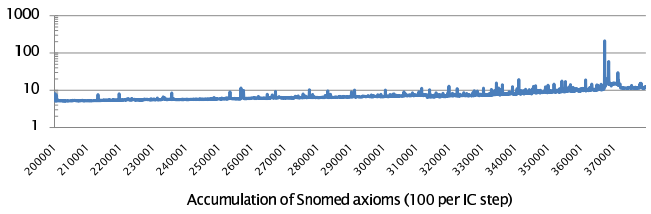


Fig. 5. Incremental classification time (second) for $\mathcal{O}^{\text{SNOMED}}$'s evolution.

phenomenon can probably be explained by the fact that concept definitions in $\mathcal{O}^{\text{NCI}}$ are somewhat large and can be broken down to a large number of smaller axioms during normalization. At any rate, the incremental classification service noticeably improved on standard classification from scratch.

The other meaningful experiment of incremental classification is to simulate the evolution of SNOMED CT. The simulation first took a subset of $\mathcal{O}^{\text{SNOMED}}$ with about two hundred thousand axioms and then classified it. After the initial classification was finished, it repeatedly supplied 100 additional axioms from the rest of $\mathcal{O}^{\text{SNOMED}}$ and incrementally classified them against the previously classified axioms. This process was carried out until no more axioms were left to be incrementally classified, i.e., the entire ontology has eventually been considered.

The time required in each step of incremental classification is plotted in Figure 5. Each of the early incremental classification steps took 6 seconds or less, and it slowly increased over the course of $\mathcal{O}^{\text{SNOMED}}$'s expansion. The median of incremental classification time over ten seconds was only 11.46 second, the amount of time that should arguably be tolerable to be adopted in realistic development environment for SNOMED CT.

*3) Subsumption query answering:* Having implemented the *goal-directed* subsumption algorithm [31], [32], CEL also supports subsumption testing directly without the need of a full classification. To evaluate this reasoning service, we have sampled[15] two sets of subsumptions as follows: *(i)* randomly select 1 000 concept names from $\mathsf{CN}(\mathcal{O})$; *(ii)* for each $A$ from step (i), sample 5 distinct *positive* subsumptions $\mathcal{O} \models A \sqsubseteq B$ with $B \notin \{A, \top\}$; *(iii)* for each $A$ from step (i), sample 5 distinct *negative* subsumptions $\mathcal{O} \not\models A \sqsubseteq B$ for some $B$. Given $\mathcal{O}$, the sets of sampled subsumptions obtained by steps (ii) and (iii) are denoted by $\mathsf{p\_subs}(\mathcal{O})$ and $\mathsf{n\_subs}(\mathcal{O})$, respectively.

Subsumptions in $\mathsf{p\_subs}(\mathcal{O})$ and $\mathsf{n\_subs}(\mathcal{O})$ were queried. The average/maximum CPU times for each subsumption query answering in each ontology are shown in Table IV. Observe that, on average, it took CEL only tiny fractions of a second to answer single subsumption queries in most cases except for the negative subsumptions in $\mathcal{O}^{\text{FULLGALEN}}$. The hardest case for the ontology took just above eight seconds in order to decide non-subsumption. Except for $\mathcal{O}^{\text{FMA}}$, the time differences between querying positive subsumptions, i.e., $\mathsf{p\_subs}(\mathcal{O})$, and negative ones, i.e., $\mathsf{n\_subs}(\mathcal{O})$, are more than double.

[15]Since there are about *144 billion pairs* of concept names in the case of $\mathcal{O}^{\text{SNOMED}}$ and some subsumption queries against $\mathcal{O}^{\text{FULLGALEN}}$ took several seconds, performing subsumption queries between *all* pairs would not be feasible; hence, the need for sampling.

| Ontology | $\mathsf{p\_subs}(\mathcal{O})$ | $\mathsf{n\_subs}(\mathcal{O})$ |
|---|---|---|
| $\mathcal{O}^{\text{GO}}$ | 0.016/10 | 0.048/50 |
| $\mathcal{O}^{\text{NCI}}$ | 0.062/10 | 0.166/10 |
| $\mathcal{O}^{\text{FMA}}$ | 0.44/10 | 0.48/10 |
| $\mathcal{O}^{\text{NOTGALEN}}$ | 0.29/10 | 0.97/20 |
| $\mathcal{O}^{\text{FULLGALEN}}$ | 75.43/7 900 | 3 477.35/8 050 |
| $\mathcal{O}^{\text{SNOMED}}$ | 0.41/20 | 1.01/1 040 |

TABLE IV
AVERAGE/MAXIMUM SUBSUMPTION TESTING TIME (MILLISECOND).

| Ontologies | Extraction time | | | |
|---|---|---|---|---|
| | median | average | maximum | total |
| $\mathcal{O}^{\text{GO}}$ | ∼0.00 | 0.0001 | 0.01 | 1.41 |
| $\mathcal{O}^{\text{NCI}}$ | ∼0.00 | 0.0001 | 0.19 | 2.19 |
| $\mathcal{O}^{\text{FMA}}$ | 0.10 | 0.0688 | 1.17 | 5 171 |
| $\mathcal{O}^{\text{NOTGALEN}}$ | ∼0.00 | 0.0005 | 0.03 | 1.42 |
| $\mathcal{O}^{\text{FULLGALEN}}$ | 0.01 | 0.0317 | 0.92 | 734 |
| $\mathcal{O}^{\text{SNOMED}}$ | ∼0.00 | 0.0082 | 5.46 | 3 110 |

TABLE V
TIME TO EXTRACT THE REACHABILITY-BASED MODULES (SECOND).

| Ontologies | Module size (%) | | |
|---|---|---|---|
| | median | average | maximum |
| $\mathcal{O}^{\text{GO}}$ | 19 (0.0928) | 28 (0.1389) | 190 (0.9284) |
| $\mathcal{O}^{\text{NCI}}$ | 12 (0.0434) | 29 (0.1048) | 436 (1.577) |
| $\mathcal{O}^{\text{FMA}}$ | 22 234 (29.59) | 14 881 (19.80) | 22 276 (29.65) |
| $\mathcal{O}^{\text{NOTGALEN}}$ | 33 (1.201) | 62 (2.250) | 435 (15.83) |
| $\mathcal{O}^{\text{FULLGALEN}}$ | 167 (0.7218) | 3 795 (16.40) | 8 553 (36.97) |
| $\mathcal{O}^{\text{SNOMED}}$ | 19 (0.0050) | 31 (0.0082) | 262 (0.0690) |

TABLE VI
SIZE OF THE REACHABILITY-BASED MODULES (♯AXIOMS AND %).

*4) Modularization:* Two sets of experiments were carried out to evaluate the modularization based on reachability [31]. In the first set, a module for each concept name in each ontology was extracted (henceforth, referred to as *c-module* for brevity). The reasons were that modules for single concepts form a good indicator of the typical size of the modules compared to the whole ontology. Moreover, modules for single concepts are especially interesting for optimization both in standard reasoning of classification and in axiom pinpointing [9], [34]. The second set of experiments concerned non-atomic signatures of varying sizes.

For each ontology $\mathcal{O}$ in the test suite and each concept name $A$ occurring in $\mathcal{O}$, the reachability-based module $\mathcal{O}_A^{\mathsf{reach}}$ was extracted. The time required to extract each c-module and its size were measured and are summarized in Table V and VI, respectively. Observe that it took only a tiny amount of time to extract a c-module based on reachability, where more than two third of all the extractions required less than 10 milliseconds. However, extracting a large number of c-modules (i.e., as many as the number of concept names) required considerably more time and even longer than classification in some cases.

Except for $\mathcal{O}^{\text{FMA}}$ and $\mathcal{O}^{\text{FULLGALEN}}$, all ontologies have relatively very small c-modules, i.e., in the range below 450 axioms. The exceptional ontologies have idiosyncratic structures, namely two distinct groups of c-modules, that were revealed by

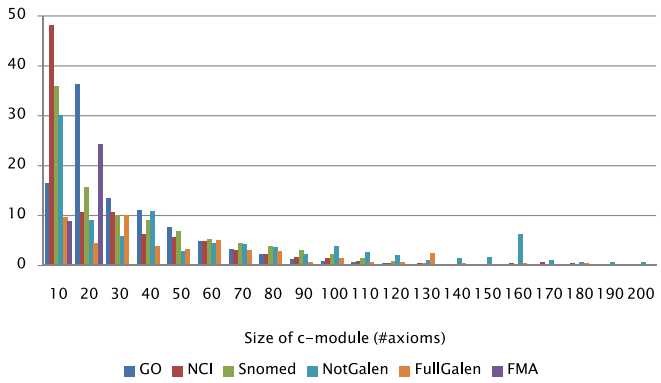


Fig. 6. Relative frequency of *small* c-modules.

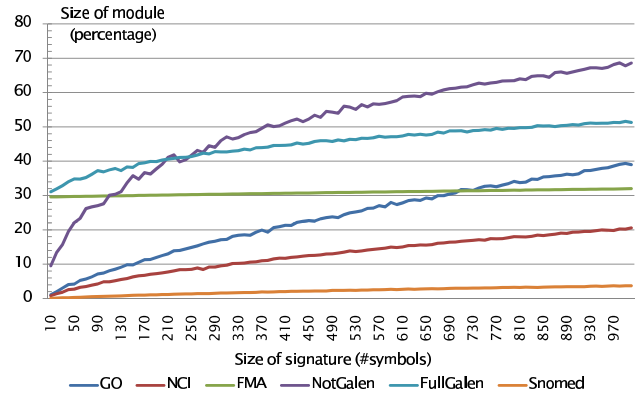


Fig. 7. Size of the reachability-based modules against size of the signature.

reachability-based modularization. In $\mathcal{O}^{\text{FullGalen}}$, just above half of all c-modules (i.e., 12 119) are of size less than 460 axioms, while the rest (i.e., 11 017) are of size between 7 875 and 8 553 axioms. Similarly, 24 867 c-modules in $\mathcal{O}^{\text{Fma}}$ are of size less than 33 axioms, and the rest of size between 22 235 and 22 276 axioms. Surprisingly, there is no c-module of size between those of these two groups. This disrupt distribution can be seen as an indicator of the presence of big cyclic dependencies in the ontologies.

The distribution of sizes of *small* c-modules in all ontologies is depicted in Figure 6.[16] The depicted distribution is natural in the sense that there are a large number of smaller c-modules and a small number of larger ones. This pattern is most vividly visible in the case of $\mathcal{O}^{\text{Go}}$, $\mathcal{O}^{\text{Nci}}$ and $\mathcal{O}^{\text{Snomed}}$, where about 50% of c-modules are of size 20 or less and about 95% of c-modules are of size 100 or less. A similar pattern can also be seen in the case of the two Galen ontologies.

To simulate ontology reuse scenario, where a part of a well-established ontology relevant to the signature of interest is imported, we have designed and performed another set of experiments. In these experiments, signatures of varying sizes from 10 to 1 000 (at 10-symbol intervals) were randomly generated from the signature of each test ontology. For each ontology $\mathcal{O}$ and each generated signature $\mathbf{S} \subseteq \mathsf{Sig}(\mathcal{O})$, the reachability-based module $\mathcal{O}^{\mathsf{reach}}_{\mathbf{S}}$ for $\mathbf{S}$ in $\mathcal{O}$ was extracted. The size of the reachability-based module is plotted against the size of the signature in Figure 7. Observe that the growth trends of $\mathcal{O}^{\text{Snomed}}$, $\mathcal{O}^{\text{Nci}}$, $\mathcal{O}^{\text{Go}}$ and $\mathcal{O}^{\text{NotGalen}}$ appear proportional to the average size of c-module in the respective ontology (c.f. Table VI). The modules in $\mathcal{O}^{\text{FullGalen}}$ and $\mathcal{O}^{\text{Fma}}$ started at a relatively large size (i.e., about 30%) because there was a good chance that one of the concept names in the signature was involved in the larger cluster of c-modules.

*5) Axiom pinpointing:* CEL supports the reasoning service of finding *a* MinA and *all* MinAs for a subsumption of interest.[17] This reasoning service was evaluated on $\mathcal{O}^{\text{Snomed}}$. As mentioned earlier, the faulty subsumption

$$\sigma : \mathsf{AmputationOfFinger} \sqsubseteq_{\mathcal{O}^{\text{Snomed}}} \mathsf{AmputationOfHand}$$

holds in $\mathcal{O}^{\text{Snomed}}$. It took CEL less than half a second to compute a MinA for $\sigma$ which turned out to be the only MinA for the subsumption. Consisting of 6 axioms, this MinA indeed indicates the culprit for the unintended subsumption [9].

This experiment was generalized to other subsumptions. However, considering all (positive) subsumptions was not feasible, since there are more than five million subsumption relationships that follow from $\mathcal{O}^{\text{Snomed}}$: assuming an average extraction time of half a second, this would have required a month. For this reason, five sets of 1 000 sampled concept names from $\mathsf{CN}(\mathcal{O}^{\text{Snomed}})$ were generated, denoted by $\mathsf{c\text{-}samples}(n)$ with $n = \{1, 2, 3, 4, 5\}$. For each $n$, the single pinpointing algorithm (to find a MinA) was run on all subsumption relationships $A \sqsubseteq_{\mathcal{O}^{\text{Snomed}}} B$ such that $A \in \mathsf{c\text{-}samples}(n)$, $B \notin \{A, \top\}$, and $\mathcal{O}^{\text{Snomed}} \models A \sqsubseteq B$. For each subsumption considered, the time to compute the module, the module's size, the time to prune its axioms to obtain a MinA and the MinA's size were measured. The average/maximum of these experimental results are listed in Table VII, segregated by $\mathsf{c\text{-}samples}(n)$.

Observe that the average time to compute a single MinA (i.e., the sum of the time results in columns 3 and 5) was less than a second in all samples. Though the extracted modules were already quite small (i.e., comprising 52 axioms on average and 165 axioms at most), the MinAs were much smaller (i.e., 7 axioms on average and 39 at most). The average size ratio is 13.41% as shown in the last column.

Among the same sampled subsumptions above, the ones with more than one MinA were considered in the evaluation of the full pinpointing algorithm (to find all MinAs and diagnoses). Based on the experiment, there were hard cases in the samples where more than 100 MinAs existed, and where it took up to 72 hours to compute all MinAs in each of these cases. For this reason, the number of computed MinAs was limited in

[16] For readability reasons, frequency bars for c-module size larger than 200 are trimmed off the chart. This does not affect the reading of the chart since more than 95% of the small c-modules are included and the ignored size values evenly disperse the trimmed area of the chart.

[17] It implemented the modularization-based pruning algorithm and hitting set tree algorithm which uses CEL per se as the subsumption reasoner [32].

| Concept samples $A$ from CN($\mathcal{O}^{\text{SNOMED}}$) | ♯Subs. samples | Time to extract module $\mathcal{O}_A^{\text{SNOMED}}$ (avg/max) | Module size (avg/max) | Pruning time for $A \sqsubseteq_{\mathcal{O}_A^{\text{SNOMED}}} B$ (avg/max) | MinA size (avg/max) | MinA/$\mathcal{O}_A^{\text{SNOMED}}$ ratio (%) |
|---|---|---|---|---|---|---|
| c-samples(1) | 14 279 | 0.0142 / 0.05 | 52.69 / 161 | 0.3122 / 7.89 | 7.08 / 37 | 13.44 |
| c-samples(2) | 14 209 | 0.0135 / 0.06 | 52.80 / 165 | 0.2226 / 8.15 | 6.91 / 35 | 13.09 |
| c-samples(3) | 14 840 | 0.0139 / 0.05 | 52.08 / 145 | 0.2201 / 8.05 | 7.13 / 39 | 13.70 |
| c-samples(4) | 14 617 | 0.0139 / 0.06 | 51.17 / 163 | 0.1790 / 3.61 | 6.87 / 35 | 13.43 |
| c-samples(5) | 14 377 | 0.0133 / 0.06 | 51.23 / 158 | 0.1828 / 3.50 | 6.86 / 35 | 13.39 |
| Overall | 72 322 | 0.0138 / 0.06 | 51.99 / 165 | 0.2231 / 8.15 | 6.97 / 39 | 13.41 |

TABLE VII
EMPIRICAL RESULTS OF THE MODULARIZATION-BASED APPROACH TO FINDING A MINA ON FIVE SETS OF SAMPLED SUBSUMPTIONS IN SNOMED CT (TIME IN SECONDS; SIZE IN NUMBER OF AXIOMS).




Fig. 8. Relative frequency of the numbers of *all* MinAs in $\mathcal{O}^{\text{SNOMED}}$.

| Samples | ♯MinAs (avg/max) | MinA size (avg/max) | $\mu$ | $\nu$ | $\mu/\nu$ |
|---|---|---|---|---|---|
| easy | 3.7 / 9 | 8.0 / 26 | 4.8 / 22 | 12.3 / 39 | 0.39 |
| hard | 10 / 10 | 16.4 / 45 | 7.0 / 30 | 32.5 / 63 | 0.22 |

TABLE VIII
STATISTICAL RESULTS ON THE COMPUTED MINAS FOR $\mathcal{O}^{\text{SNOMED}}$.

our experiment to 10. Therefore, the statistics shown in the following will be divided into two groups:

- easy-samples comprising 2–9 MinAs, and
- hard-samples comprising at least 10 MinAs.

Based on all the subsumptions considered, 10 492 (56.19%) subsumptions belong to easy-samples, and 8 181 (43.81%) subsumptions to hard-samples.

Table VIII shows the average/maximum numbers of MinAs (♯MinAs) and their size. It also presents the average/maximum numbers of common axioms in all MinAs, i.e., $\mu = |\bigcap_{\text{MinAs } \mathcal{S} \text{ for } \sigma} \mathcal{S}|$, and those of all axioms in all MinAs, i.e., $\nu = |\bigcup_{\text{MinAs } \mathcal{S} \text{ for } \sigma} \mathcal{S}|$. The average ratio $\mu/\nu$, which indicates the degree of commonality of the computed MinAs, is shown in the last column of the table. The statistical results for easy-samples are complete w.r.t. all the MinAs, whereas those for hard-samples are partial. The relative distribution of ♯MinAs below ten is shown in Figure 8. More than half of all the considered subsumptions (51.51%) have 7 MinAs or less, i.e., the median of ♯MinAs for easy-samples and hard-samples collectively is 7. Though little can be said about the distribution of ♯MinAs larger than 9, it is known from the test results that about 43% have ten or more MinAs and that the largest known ♯MinAs is 158. It can be observed from the table that the MinA size is larger when there are more MinAs, i.e., a MinA for easy-samples is of size 8 axioms on average, whereas a MinA for hard-samples is of size 16. Interestingly, the degree of commonality of the axioms in all MinAs is quite high, i.e., $\mu/\nu$ is 0.39 and 0.22 for the easy and hard cases, respectively. This means that about one third of axioms are shared among all the MinAs.

On average, it took 8.88 (37.88, resp.) seconds and required 178 (770, resp.) subsumption calls to compute all MinAs for easy-samples (10 MinAs for hard-samples, resp.) Again, it can be argued that this time is tolerable to be adopted in realistic development environment for SNOMED CT, especially considering the facts that the method generates MinAs one after the other and that the first MinA becomes available in

less than half a second.

## VI. Conclusion

This paper investigated a number of life science ontologies and identified their common characteristics that can mostly be met by the lightweight DL $\mathcal{EL}^+$. It also presented promising empirical results of various DL-based reasoning services using those ontologies as benchmarks. With these investigation and empirical results, it is possible to claim that the use of the lightweight DL is beneficial both in terms of expressivity—i.e., it is sufficiently expressive to formulate most biomedical ontologies—and scalability—i.e., tractable reasoning allows to deal with large-scale ontologies in a robust manner.

## Acknowledgment

The author would like to thank Franz Baader and Carsten Lutz for their valuable advice and countless discussions. This work was partially supported by the DFG project under grant BA1122/11-1 and the EU project TONES.

## References

[1] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The gene ontology consortium. *Nature Genetics*, 25(1):25–29, May 2000.

[2] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI-05*, Edinburgh, UK, 2005. Morgan-Kaufmann Publishers.

[3] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2007.

[4] F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL—a polynomial-time reasoner for life science ontologies. In *Proc. IJCAR'06*, volume 4130 of LNAI, pages 287–291. Springer, 2006.

[5] F. Baader. Terminological cycles in a description logic with existential restrictions. In *Proc. of the IJCAI-03*, pages 325–330. Morgan Kaufmann, 2003.

[6] F. Baader, B. Hollunder, B. Nebel, H.-J. Profitlich, and Enrico Franconi. An emperical analysis of optimization techniques for terminological representation systems or: "making kris get a move on". *Applied Intelligence*, 4(2):109–132, 1994.

[7] F. Baader, C. Lutz, and B. Suntisrivaraporn. Is tractable reasoning in extensions of the description logic $\mathcal{EL}$ useful in practice? In *Proc. of M4M-05*, Berlin, Germany, 2005.

[8] F. Baader, R. Peñaloza, and B. Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In *Proc. of the German AI Conference (KI-07)*, LNAI, Osnabrück, Germany, 2007. Springer.

[9] F. Baader and B. Suntisrivaraporn. Debugging SNOMED CT using axiom pinpointing in the description logic $\mathcal{EL}^+$. In *Proceedings of the 3rd Knowledge Representation in Medicine Conference (KR-MED'08): Representing and Sharing Knowledge Using SNOMED*, Phoenix AZ, USA, 2008.

[10] R. J. Brachman and H. J. Levesque. The tractability of subsumption in frame-based description languages. In *Proceedings of the 4th Nat. Conf. on Artificial Intelligence (AAAI'84)*, pages 34–37, 1984.

[11] S. Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In *Proc. of ECAI-04*, pages 298–302. IOS Press, 2004.

[12] C. Golbreic, M. Horridge, I. Horrocks, B. Motik, and R. Shearer. OBO and OWL: Leveraging semantic web technologies for the life sciences. In *Proc. of ISWC-07*, volume 4825 of LNCS, pages 169–182. Springer, 2007.

[13] T. R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.

[14] V. Haarslev and R. Möller. Racer system description. In *Proc. of IJCAR-01*, pages 701–705, Siena, Italy, 2001. Springer-Verlag.

[15] V. Haarslev, R. Möller, and S. Wandelt. The revival of structural subsumption in tableau-based description logic reasoners. In *Proc. of DL-08*, CEUR-WS, 2008.

[16] I. Horrocks. *Optimising Tableaux Decision Procedures for Description Logics*. PhD thesis, University of Manchester, 1997.

[17] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. of Web Semantics*, 1(1):7–26, 2003.

[18] I. Horrocks and D. Tsarkov. Optimised classification for taxonomic knowledge bases. In *Proc. of DL-05*, pages 184–191, 2005.

[19] I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible $\mathcal{SROIQ}$. In *Proc. of KR-06*, pages 57–67. AAAI Press, 2006.

[20] I. Horrocks, A. Rector, and C. A. Goble. A description logic based schema for the classification of medical data. In *Knowledge Representation Meets Databases, Proc. of the 3rd Workshop KRDB'96, Budapest, Hungary, August 13, 1996*, 1996.

[21] C. Lutz, F. Baader, E. Franconi, D. Lembo, R. Möller, R. Rosati, U. Sattler, B. Suntisrivaraporn, and S. Tessaris. Reasoning support for ontology design. In Bernardo Cuenca Grau, Pascal Hitzler, Connor Shankey, and Evan Wallace, editors, *Proc. of OWLED-06*, 2006.

[22] B. Motik. *Reasoning in Description Logics using Resolution and Deductive Databases*. PhD thesis, Univesität Karlsruhe (TH), Karlsruhe, Germany, January 2006.

[23] B. Motik, R. Shearer, and I. Horrocks. Optimized Reasoning in Description Logics using Hypertableaux. In Frank Pfenning, editor, *Proc. of CADE-07*, volume 4603 of *LNAI*, pages 67–83, Bremen, Germany, July 17–20 2007. Springer.

[24] B. Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43:235–249, 1990.

[25] M. O'Neil, C. Payne, and J. Read. Read codes version 3: A user led terminology. *Methods of Information in Medicine*, 34:187-921, 1995.

[26] P. Patel-Schneider and B. Swartout. Description-logic knowledge representation system specification from the krss group of the arpa knowledge sharing effort. Technical report, DARPA Knowledge Representation System Specification (KRSS) Group of the Knowledge Sharing Initiative, 1993.

[27] A. Rector and I. Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proc. of the Workshop on Ontological Engineering, AAAI Spring Symposium (AAAI'97)*, Stanford, CA, 1997. AAAI Press.

[28] K. A. Spackman, K. E. Campbell, and R. A. Cote. SNOMED RT: A reference terminology for health care. In *Proc. of the 1997 AMIA Annual Fall Symposium*, pages 640–644. Hanley&Belfus, 1997.

[29] M. Q. Stearns, C. Price, K.A. Spackman, and A.Y. Wang. SNOMED clinical terms: overview of the development process and project status. In *Proc. of the 2001 AMIA Annual Symposium*, pages 662–666. Hanley&Belfus, 2001.

[30] B. Suntisrivaraporn. Optimization and implementation of subsumption algorithms for the description logic $\mathcal{EL}$ with cyclic TBoxes and general concept inclusion axioms. Master thesis, TU Dresden, Germany, 2005.

[31] B. Suntisrivaraporn. Module extraction and incremental classification: A pragmatic approach for $\mathcal{EL}^+$ ontologies. In Sean Bechhofer, Manfred Hauswirth, Joerg Hoffmann, and Manolis Koubarakis, editors, *Proc. ESWC-08*, volume 5021 of LNCS, pages 230–244. Springer-Verlag, 2008.

[32] B. Suntisrivaraporn. *Polynomial-Time Reasoning Support for Design and Maintenance of Large-Scale Biomedical Ontologies*. PhD thesis, TU Dresden, Institute for Theoretical Computer Science, Germany, 2008. Submitted for review.

[33] B. Suntisrivaraporn, F. Baader, S. Schulz, and K. Spackman. Replacing SEP-triplets in SNOMED CT using tractable description logic operators. In Jim Hunter Riccardo Bellazzi, Ameen Abu-Hanna, editor, *Proc. of AIME-07*, volume 4594 of LNCS, pages 287–291. Springer, 2007.

[34] B. Suntisrivaraporn, G. Qi, Q. Ji, and P. Haase. A modularization-based approach to finding all justifications for OWL DL entailments. In John Domingue and Chutiporn Anutariya, editors, *Proc. of ASWC-08*, LNCS. Springer, 2008.

[35] Q. H. Vu. Subsumption in the description logic in $\mathcal{ELHI}f_{R^+}$ w.r.t. general tboxes. Master thesis, TU Dresden, Germany, 2008.