Structured Objects in OWL: Representation and Reasoning

Boris Motik
University of Oxford
Oxford, UK

Bernardo Cuenca Grau
University of Oxford
Oxford, UK

Ulrike Sattler
University of Manchester
Manchester, UK

ABSTRACT

Applications of semantic technologies often require the representation of and reasoning with *structured objects*—that is, objects composed of parts connected in complex ways. Although OWL is a general and powerful language, its class descriptions and axioms cannot be used to describe arbitrarily connected structures. An OWL representation of structured objects can thus be underconstrained, which reduces the inferences that can be drawn and causes performance problems in reasoning. To address these problems, we extend OWL with *description graphs*, which allow for the description of structured objects in a simple and precise way. To represent conditional aspects of the domain, we also allow for SWRL-like rules over description graphs. Based on an observation about the nature of structured objects, we ensure decidability of our formalism. We also present a hypertableau-based decision procedure, which we implemented in the HermiT reasoner. To evaluate its performance, we have extracted description graphs from the GALEN and FMA ontologies, classified them successfully, and even detected a modeling error in GALEN.

Categories and Subject Descriptors

I.2.4 [**KR Formalisms and Methods**]: OWL

General Terms

Theory, Languages

1. INTRODUCTION

Ontologies are nowadays used in disciplines as diverse as biology [20], medicine[18], astronomy [4], and agriculture [21]. A de facto standard for ontology modeling is the Web Ontology Language (OWL),[1] so most ontologies in these domains were either developed from the start using OWL or translated into OWL from other formalisms. OWL is an expressive language capable of supporting diverse applications. Its logical underpinning is given by description logics (DLs), which provide OWL with a clean model-theoretic semantics, well-understood reasoning problems, and powerful reasoners.

[1]In this paper, we focus on OWL DL—the most expressive of the decidable languages of the OWL family.

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

Structured objects—that is, objects composed of other, possibly interrelated objects—pose some well-known problems to OWL and DLs [3, 1, 19]. Such objects abound, for example, in molecular biology and the clinical sciences. Clinical ontologies such as GALEN [22], the Foundation Model of Anatomy (FMA) [18], the National Cancer Institute (NCI) Thesaurus [7], and SNOMED CT [23] are currently being used in large-scale applications, and they describe numerous structured objects. For example, GALEN models the heart as consisting of the left and the right ventricles, the two atria, and the valves, all of which participate in complex relationships, such as "the two ventricles of a heart are separated by the intraventricular septum."

OWL can be used to describe domains consisting of an arbitrary or even infinite number of objects, but it only allows for axioms that can connect these objects in a certain tree-like manner. In other words, OWL enjoys (a variant of) the *tree model property* [24]: if an OWL ontology has a model, then it has a model with a tree-like (or forest-like) relational structure as well. This property is responsible for the decidability of OWL reasoning [24]; however, it prevents sufficiently accurate description of complex structured objects, since schema-level axioms in OWL cannot describe arbitrary relational structures. Consider the previously mentioned diamond-shaped structure involving a heart, its right and left ventricles, and a septum. In addition to a model that corresponds to the structure in which the objects are connected as expected, each schema-level description of the heart in OWL will also have a model where one heart has two septa, each as a part of the left and the right ventricle, respectively. Thus, certain consequences of the diamond-shaped structure cannot be drawn from its formulation in OWL. For example, we cannot conclude that, if the right ventricle has a perforated septum, the left ventricle has a perforated septum as well.

To address this lack of expressive power, in Section 4 we propose an extension of OWL for modeling structured objects using *description graphs*. Such graphs consist of vertices labeled with atomic concepts and edges labeled with atomic roles. According to our proposed model-theoretic semantics, these graphs are class-level statements that specify general patterns of connections between objects. In addition, we allow for SWRL-like rules [8] to enable the description of conditional statements about graphs.

Extending DLs with axioms that can enforce arbitrary structures easily leads to undecidability [12]. Our formalism, however, is decidable because it can represent only structured objects whose number of parts is bounded. In prac-

tice, structured objects are usually modeled up to a certain level of granularity, which naturally determines this bound. In Section 5, we present a reasoning algorithm for the case where the OWL part is expressed in $\mathcal{SHIQ}$ [10]; it should, however, be possible to extended the algorithm to $\mathcal{SHOIQ}$ [9] and hence cover OWL DL. We thus obtain a powerful, decidable, and practicable language that combines two complementary formalisms: unbounded but tree-like structures can be described using standard OWL axioms, and the naturally bounded structured parts can be described using arbitrarily connected description graphs and rules.

We have implemented our algorithm in the DL reasoner HermiT [16].[2] The validation of our approach is currently difficult due to the lack of test data. Thus, we have devised an algorithm that extracts description graphs from OWL ontologies, and have applied it to GALEN and FMA. The resulting ontologies should be treated with caution; however, domain experts have confirmed that substantial parts of the ontologies reflect the actual human anatomy. Our transformation can thus be a starting point for a more comprehensive remodeling using description graphs. Finally, the ontologies are sufficiently complex to allow us to estimate the practicability of reasoning. We present the transformation algorithm in Section 6.

In Section 7, we discuss the results obtained by classifying the transformed ontologies. Our transformation allowed us to discover a modeling error in GALEN, which we take as indication that our formalism can indeed be useful in practice. Furthermore, classification times for the transformed ontologies are of similar orders of magnitude as for the original ontologies despite the fact that our formalism adds considerable expressive power to OWL.

Due to lack of space, proofs and certain technical details are included in the accompanying technical report [13].

2. PRELIMINARIES

A $\mathcal{SHIQ}$ signature is a triple $\Sigma = (N_C, N_R, N_I)$ consisting of mutually disjoint sets of *atomic concepts* N_C, *atomic roles* N_R, and *individuals* N_I. Let S be an atomic role, A an atomic concept, and n a nonnegative integer. $\mathcal{SHIQ}$ *roles* and *concepts* are defined using the following grammar:

$$R \to S \mid S^-$$
$$C \to \top \mid \bot \mid A \mid \neg C \mid C_1 \sqcup C_2 \mid C_1 \sqcap C_2 \mid \forall R.C \mid \exists R.C \mid \geq n\, R.C \mid \leq n\, R.C$$

A *TBox* $\mathcal{T}$ is a finite set of *role inclusions* $R_1 \sqsubseteq R_2$, *transitivity axioms* $\mathsf{Trans}(R)$, and *general concept inclusions* (GCIs) $C_1 \sqsubseteq C_2$. An *extensionally reduced ABox* $\mathcal{A}$ is a finite set of assertions $(\neg)A(a)$, $S(a,b)$, $a \approx b$, and $a \not\approx b$. A knowledge base $\mathcal{K}$ is a pair $(\mathcal{T}, \mathcal{A})$. To ensure decidability of reasoning, certain restrictions apply on the usage of roles in at-most and at-least concepts $\leq n\, R.C$ and $\geq n\, R.C$ [10]. An *interpretation* I is a first-order relational structure. The definition of satisfaction of axioms in I can be found in [10]. I is a *model* of $\mathcal{K}$, written $I \models \mathcal{K}$, if it satisfies all axioms of $\mathcal{K}$. The basic inference problem for $\mathcal{SHIQ}$ is checking satisfiability of $\mathcal{K}$—that is, checking whether a model of $\mathcal{K}$ exists.

The principles for extending DLs with rules can be found in [12, 5, 8]. For a $\mathcal{SHIQ}$ signature Σ, let N_V be a countably infinite set of *variables* disjoint from N_I. A *predicate* is a concept, a role, or the *equality predicate* $\approx$. Concepts have arity one, and roles and $\approx$ have arity two. An *atom* over Σ has the form $P(x_1, \ldots, x_n)$, where P is a predicate of arity n and x_i are variables. Atoms with the equality predicate are written as $t_1 \approx t_2$. A rule r is an expression of the form

$$B_1 \wedge \ldots \wedge B_n \to H_1 \vee \ldots \vee H_m \tag{1}$$

where $n \geq 0$, $m \geq 0$, and B_i and H_j are atoms. The set of atoms $\{B_1, \ldots, B_n\}$ is called the *antecedent*, and the set of atoms $\{H_1, \ldots, H_m\}$ is called the *consequent*. A *program* $\mathcal{P}$ is a finite set of rules. A rule r is interpreted in an interpretation I as a standard first-order implication.

3. MOTIVATION

To understand the limitations of modeling structured objects in OWL, let us consider modeling the anatomy of the heart shown in Figure 1(a). This example has been derived by reconstructing the intention behind the axioms describing the heart in GALEN. We next consider possibilities for a logical interpretation of the figure.

Figure 1(a) could be represented in OWL using an ABox $\mathcal{A}$. ABox assertions, however, represent concrete data; thus, $\mathcal{A}$ would represent the structure of *one particular* heart. In this paper, we are concerned with modeling structured objects *at the schema level*—that is, we want to describe the general structure of *all* hearts. We should be able to instantiate such a description many times. For example, if we say that each patient has a heart, then, for each concrete patient, we should instantiate a *different* heart, each of the structure shown in Figure 1(a). This clearly cannot be achieved if we describe the structure of the heart using ABox assertions. Consequently, GALEN, SNOMED CT, and NCI contain only schema-level axioms and no ABox assertions.

We can give a logical, schema-level interpretation to Figure 1(a) by treating vertices as concepts and arrows as *participation constraints* specifying relationships between concepts. For example, *LeftSideOfHeart* and *AorticValve* are concepts and the arrow between them states that each left side of the heart has an aortic valve as a structural component. Participation constraints can be represented using existential quantification, which can be encoded in OWL using axioms of the form (2). Let $\mathcal{K}$ be a DL knowledge base containing the following axioms.

$$\mathit{LeftSideOfHeart} \sqsubseteq \exists \mathit{hasStructuralComponent}.\mathit{AorticValve} \tag{2}$$
$$\mathit{AorticValve} \sqsubseteq \exists \mathit{hasAlphaConnection}.\mathit{LeftVentricle} \tag{3}$$
$$\mathit{LeftSideOfHeart} \sqsubseteq \exists \mathit{hasSolidDivision}.\mathit{LeftVentricle} \tag{4}$$

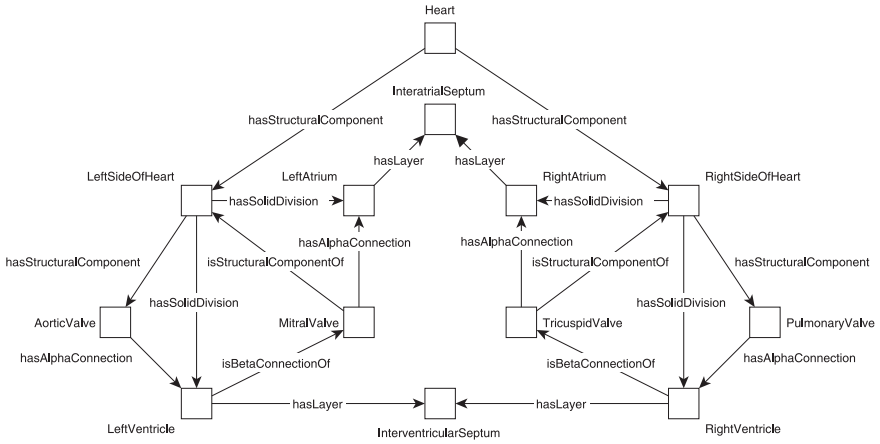
Let I be an interpretation that corresponds to Figure 1(a) in the obvious way. Clearly, I is a model of $\mathcal{K}$, which justifies the formalization of Figure 1(a) by axioms (2)–(4).

Such a schema-level representation of a heart can be put to use in many ways. We might represent knowledge about various heart conditions; for example, if the aortic valve suffers from aortic regurgitation (AR), then the left ventricle suffers from left ventricular hypertrophy (LVH):

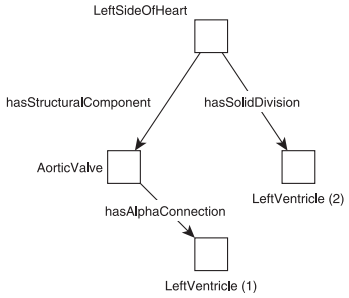
$$\mathit{AorticValve} \sqcap \mathit{HasAR} \sqsubseteq \forall \mathit{hasAlphaConnection}.\mathit{HasLVH} \tag{5}$$

We might expect to derive from (2)–(5) that, if the aortic valve of the left side of the heart suffers from aortic regur-

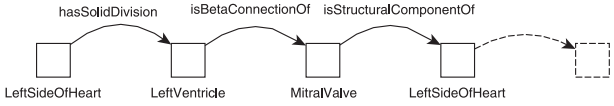
[2] http://web.comlab.ox.ac.uk/oucl/work/boris.motik/HermiT/



(a) Intended Structure



(b) Unintended Tree Structure



(c) Unintended Infinite Structure

Figure 1: Different Models of the Heart in GALEN

gitation, then the left ventricle suffers from hypertrophy:

$$\begin{aligned} &\mathit{LeftSideOfHeart} \sqcap \\ &\exists \mathit{hasStructuralComponent}.(\mathit{AorticValve} \sqcap \mathit{HasAR}) \\ &\qquad \sqsubseteq \exists \mathit{hasSolidDivision}.\mathit{HasLVH} \end{aligned} \tag{6}$$

Unfortunately, (6) does not follow from $\mathcal{K}$: axioms (3) and (4) imply the existence of two left ventricles, but no axiom in $\mathcal{K}$ states that these two ventricles are necessarily the same object. Thus, an interpretation I' corresponding to Figure 1(b) is also a model of $\mathcal{K}$. In I', even if the aortic valve has aortic regurgitation, the second left ventricle is unaffected. Hence, $I' \not\models (6)$, so $\mathcal{K} \not\models (6)$ as well.

The knowledge base $\mathcal{K}$ is thus underconstrained: some models of $\mathcal{K}$ do not correspond to the actual structure of the heart shown in Figure 1(a). This discrepancy can prevent us from drawing some quite reasonable conclusions, such as (6). Furthermore, it can also cause problems with the performance of reasoning. For example, we might use axioms (4) and (7)–(8) to describe the relationships between the left side of the heart, the left ventricle, and the mitral valve.

$$\mathit{LeftVentricle} \sqsubseteq \exists \mathit{isBetaConnectionOf}.\mathit{MitralValve} \tag{7}$$

$$\begin{aligned} &\mathit{MitralValve} \sqsubseteq \\ &\qquad \exists \mathit{isStructuralComponentOf}.\mathit{LeftSideOfHeart} \end{aligned} \tag{8}$$

While admitting a model corresponding to Figure 1(a), these axioms do not state that the mitral valve in (7) is a structural component of the "initial" left side of the heart. Hence, the interpretation from Figure 1(c) is also a model of these axioms. In fact, the latter model is "canonical" in the sense that it reflects the least amount of information derivable from the axioms. In order to disprove an entailment from these axioms, an OWL reasoner will try to construct such a "canonical" model. In practice, such models can be highly repetitive and much larger than the intended models, which, according to our experience, is the main reason why OWL reasoners still cannot process ontologies such as FMA and certain versions of GALEN.

To avoid such problems, we need to extend $\mathcal{K}$ with additional axioms that make *all* models of $\mathcal{K}$ correspond as much as possible to the intended conceptualization shown in Figure 1(a). Such axioms, however, cannot be stated in OWL, for reasons we explain next. OWL can represent *unbounded* or even *infinite* domains, which is appropriate in many cases. For example, in the domain of people, we should not make any assumptions about the number of people in the world. In other words, the domain of all people does not exhibit a *natural bound* on its size. Thus, we can represent the fact that every person has exactly two parents who are persons:

$$\mathit{Person} \sqsubseteq \geq 2\, \mathit{hasParent}.\mathit{Person} \sqcap \leq 2\, \mathit{hasParent}.\top \tag{9}$$

Reasoning with such axioms is not straightforward. A model containing one person γ must contain two parents δ_1 and δ_2, each of which requires the existence of two additional parents and so on. Effectively, we obtain a model that is similar to the one shown in Figure 1(c).

To ensure termination of the model construction outlined in the previous paragraph, the structure of the axioms allowed in OWL is restricted such that the language exhibits (a variant of) the *tree model property* [24]: whenever a knowledge base $\mathcal{K}$ has a model, it also has a model of a certain tree shape. The relationship between the left side of the heart, the aortic valve, and the left ventricle in Figure 1(a) is, however, triangular and cannot be represented as a tree. Hence, if we want to ensure that the ventricles whose existence is implied by (3) and (4) are the same in *every* model of $\mathcal{K}$, we must leave the confines of OWL and DLs.

Certain rule formalisms can axiomatize nontree structures.

For example, the following SWRL [8] rule can be used to make the two ventricles from Figure 1(b) the same:

$$\begin{array}{l} \mathit{LeftSideOfHeart}(x) \wedge \\ \quad \mathit{hasStructuralComponent}(x,y) \wedge \\ \quad \mathit{hasAlphaConnection}(y,z) \wedge \mathit{LeftVentricle}(z) \wedge \\ \quad \mathit{hasSolidDivsion}(x,w) \wedge \mathit{LeftVentricle}(w) \rightarrow z \approx w \end{array} \tag{10}$$

This, however, has significant drawbacks. From the standpoint of modeling, such a solution is quite complex, as it requires the modeler to anticipate which objects need to be made the same. The fact that the two left ventricles are the same follows from the complex interaction between axioms (2)–(4) and (10), and is thus not represented explicitly. Clearly, such a modeling formalism is likely to be hard to use and susceptible to modeling errors. From the standpoint of automated reasoning, the extension of OWL with SWRL is undecidable [8], which is a significant impediment to the adoption of SWRL in practice.

SWRL-like rules can, however, naturally express certain conditional aspects of structured objects. For example, if the septum has a ventricular septal defect, then there is a blood flow from the left to the right ventricle:

$$\begin{array}{l} \mathit{IntraventricularSeptum}(x) \wedge \mathit{HasVSD}(x) \wedge \\ \quad \mathit{hasLayer}(y_1,x) \wedge \mathit{LeftVentricle}(y_1) \wedge \\ \quad \mathit{hasLayer}(y_2,x) \wedge \mathit{RightVentricle}(y_2) \rightarrow \\ \qquad\qquad\qquad \mathit{hasBloodFlow}(y_1,y_2) \end{array} \tag{11}$$

The variables in the antecedent of this rule are connected in a non-tree-like way, so such a rule cannot be expressed in OWL. If we, however, deal with arbitrarily connected structures, such as the one shown in Figure 1(a), non-tree-like antecedents are essential for drawing the correct inferences.

Various decidable combinations of DLs and rules cannot be used for schema modeling. For example, the DL-safe rules [15] are syntactically restricted such that they apply only to the explicitly named objects. Role-safe [12] and weakly safe [17] rules also impose restrictions that prevent the application of the rules to arbitrary elements of the domain, and similar restrictions are also employed by various nonmonotonic rule extensions of OWL [6, 17, 14]. While these are quite useful in query answering, they cannot be used to derive new conclusions from the schema.

The DL $\mathcal{SROIQ}$ [11] and the OWL 1.1 extension of OWL DL extend OWL with *complex role inclusions* of the form $R_1 \circ \ldots \circ R_n \sqsubseteq S$, restricted appropriately to ensure decidability. Such axiom solve some of the problems; however, they still cannot axiomatize arbitrary structures such as the one in Figure 1(a) or express axioms such as (11).

4. DESCRIPTION GRAPHS

We now present an extension of OWL that addresses the problems from Section 3.

4.1 Basic Principles

The main aspect of a description of a structured object is the connection between the object's parts, which can naturally be represented as a graph. Hence, we introduce the notion of a *description graph* $G = (V, E, \lambda)$—a directed graph in which each vertex $i \in V$ is labeled with a set of atomic concepts $\lambda\langle i\rangle$ and each edge $\langle i, j\rangle \in E$ is labeled with a set of atomic roles $\lambda\langle i, j\rangle$. For example, Figure 1(a) can be understood as a description graph that describes the heart.

Semantically, $G = (V, E, \lambda)$ should be understood as a "template" for a fragment of a model. Let I be a model and A an atomic concept labeling some graph vertex $i \in V$. If I contains an object γ such that $\gamma \in A^I$, then I must also contain an instance of G in which γ corresponds to i. For example, if I contains an instance γ of the *Heart* concept, then I must contain a relational structure corresponding to Figure 1(a) in which γ corresponds to the top-most vertex.

As discussed in Section 3, extending DLs with constructs that allow the description of arbitrarily connected structures of unbounded size easily leads to undecidability. In practice, structured objects are usually modeled up to a certain level of granularity, which naturally determines this bound. For example, a human body consists of a certain number of organs. These organs might be decomposed into smaller parts; however, each such decomposition is bounded, so the entire model of human anatomy requires a bounded number of objects. Even though the number of required objects may be large and difficult to determine by hand, the fact that the domain is bounded is intrinsic to the modeling problem. The reasoning algorithm presented in Section 5 uses this bound to ensure termination even on arbitrarily connected, non-tree-like structures.

We assume that the set of atomic roles is divided into a set of *atomic tree roles* N_{R_t} and a set of *atomic graph roles* N_{R_g}. A *graph-extended DL knowledge base* is a 4-tuple $\mathcal{K} = (\mathcal{T}, G, \mathcal{P}, \mathcal{A})$ where $\mathcal{T}$ is a DL TBox, G is a description graph, $\mathcal{P}$ is a set of rules, and $\mathcal{A}$ is an ABox. Furthermore, $\mathcal{T}$ is allowed to refer only to tree roles, G and $\mathcal{P}$ are allowed to refer only to the graph roles, and $\mathcal{A}$ is allowed to refer to both graph and tree roles.

For example, let $\mathcal{K} = (\mathcal{T}, G, \mathcal{P}, \mathcal{A})$ be a graph-extended DL knowledge base with the following components. Let $\mathcal{T}$ contain the axioms (12)–(14). Intuitively, axiom (12) says that each person has a parent and a heart; axiom (13) ensures that the heart of each sufferer from aortic regurgitation is an instance of *HasAR*; and axiom (14) says that, on each aortic valve suffering from aortic regurgitation, some person is performing a surgery on it.

$$\mathit{Person} \sqsubseteq \exists \mathit{hasParent}.\mathit{Person} \sqcap \exists \mathit{hasHeart}.\mathit{Heart} \tag{12}$$

$$\mathit{AR\_Sufferer} \sqsubseteq \forall \mathit{hasHeart}.\mathit{HasAR} \tag{13}$$

$$\mathit{AorticValve} \sqcap \mathit{HasAR} \sqsubseteq \exists \mathit{performsSurgeryOn}^{-}.\mathit{Person} \tag{14}$$

Let G correspond to Figure 1(a), let $\mathcal{P}$ contain the rule (15) that propagates the *HasAR* concept over the structural components of the heart, and let $\mathcal{A}$ contain the assertions $\mathit{Person}(a)$ and $\mathit{AR\_Sufferer}(a)$.

$$\mathit{HasAR}(x) \wedge \mathit{hasStructuralComponent}(x,y) \rightarrow \mathit{HasAR}(y) \tag{15}$$

The semantics of graph-extended KBs ensures that each model I of $\mathcal{K}$ is of the form shown in Figure 2. I consists of two distinct parts. The *tree backbone* consists of objects (shown as large squares) connected through tree roles (shown using thick lines), and it is constructed using the standard DL axioms in $\mathcal{T}$. As discussed in Section (3), the number of persons is not naturally bounded so, if we want a decidable formalism, we must employ standard DL restrictions. Apart from the tree backbone, I also contains arbitrarily connected but naturally bounded *graph instances*, such as the structure of the heart of each person. Unlike

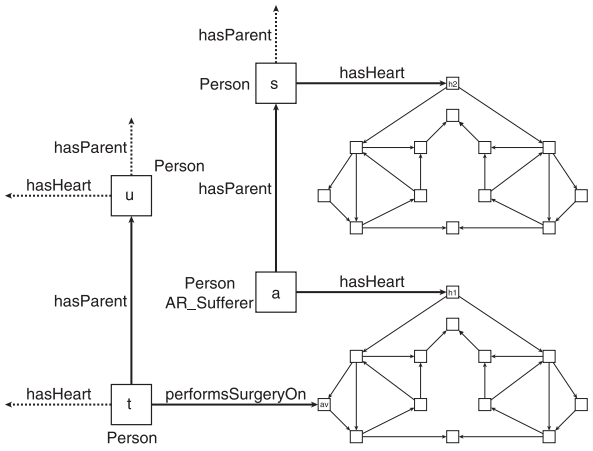


Figure 2: A Typical Model of $\mathcal{K}$

in the case of axioms (2)–(4) and Figure 1(b), each graph instance is *necessarily* of the form as specified by G in each such model I. Note that the tree backbone of I need not be contiguous: the bottom-most *AorticValve* object av can be connected to other objects through tree roles. To summarize, for a graph-extended knowledge base $\mathcal{K}$, we can consider only models that consist of graph instances, connected among themselves and with other objects through tree roles.

Decidability of the formalism is now ensured by the separation of the roles into tree and graph ones. The axioms in $\mathcal{T}$ can propagate constraints across tree roles just like in standard DLs; however, we can adapt the blocking technique [10] to ensure termination of model construction. Furthermore, the rules in $\mathcal{P}$ can propagate constraints within a graph; however, the size of the graph is naturally bounded, so this does not cause termination problems either.

Our way of obtaining decidability is related to fusions of abstract description systems (ADSs) [2], which allow for the combination of different modal and description logics. The component ADSs can share concepts; however, the interaction between them through roles is restricted to ensure decidability. Our separation of roles into graph and tree ones is similar in spirit. Bounded structures and rules, however, cannot directly be expressed as an ADS. In addition, we present a practical decision procedure.

4.2 Formalization

We now define of our language formally. We start by defining a signature that separates tree from graph roles. All subsequent definitions in this paper are implicitly parameterized with such a signature.

DEFINITION 1. *A* graph-extended DL signature *is a 4-tuple* $\Sigma = (N_C, N_{R_t}, N_{R_g}, N_I)$ *consisting of pair-wise disjoint sets of atomic concepts* N_C, atomic tree roles N_{R_t}, atomic graph roles N_{R_g}, *and individuals* N_I.

We now define description graphs formally. We make the technical assumption that the vertices of G are consecutive integers, as this allows us to use vertices as indices.

DEFINITION 2. *A* description graph $G = (V, E, \lambda)$ *is a directed labeled graph where (*i*)* $V = \{1, \ldots, \ell\}$ *is a finite set of integers called* vertices, *(*ii*)* $E \subseteq V \times V$ *is a set of* edges, *and (*iii*)* λ *labels each vertex* $i \in V$ *with a set of atomic concepts* $\lambda\langle i\rangle \subseteq N_C$, *and each edge* $\langle i, j\rangle \in E$ *with a set of atomic graph roles* $\lambda\langle i, j\rangle \subseteq N_{R_g}$. *For an atomic concept* A, V_A *is the set of vertices that contain* A *in their label:*

$$V_A = \{k \in V \mid A \in \lambda\langle k\rangle\}.$$

We now define the notion of graph-extended DL knowledge bases. The definition of graph-regular rules ensures that each such rule can become applicable only to objects from the same instance of the description graph G, which is required to obtain a decidable formalism.

DEFINITION 3. *A rule of the form* (1) *is* graph-regular *if it uses only atomic concepts and graph roles and, for each pair of variables* x_1 *and* x_2 *occurring in* r, *some antecedent atom of* r *contains both* x_1 *and* x_2.

A graph-extended DL knowledge base $\mathcal{K} = (\mathcal{T}, G, \mathcal{P}, \mathcal{A})$ *is a 4-tuple where (*i*)* $\mathcal{T}$ *is DL TBox over the signature* (N_C, N_{R_t}, N_I), *(*ii*)* G *is a description graph with* ℓ *vertices, (*iii*)* $\mathcal{P}$ *is a finite set of graph-regular rules, and (*iv*)* $\mathcal{A}$ *is an extensionally-reduced ABox over* $(N_C, N_{R_t}, N_{R_g}, N_I)$ *that, apart from standard assertions, can also contain* graph assertions *of the form* $G(a_1, \ldots, a_\ell)$ *for* $a_i \in N_I$.

Graph-regular rules can express conjunctive queries over G, so we do not consider query answering separately. We now formalize the semantics of description graphs.

DEFINITION 4. *An interpretation* $I = (\triangle^I, \cdot^I)$ *interprets a description graph* $G = (V, E, \lambda)$ *with* ℓ *vertices as an* ℓ*-ary relation* $G^I \subseteq (\triangle^I)^\ell$. *An interpretation* I *satisfies* G, *written* $I \models G$, *if all of the following conditions hold.*

***i*-key property**: *for each* $1 \le i \le \ell$,

$$\forall x_1, \ldots, x_\ell, y_1, \ldots, y_\ell \in \triangle^I : \langle x_1, \ldots, x_\ell\rangle \in G^I \wedge \langle y_1, \ldots, y_\ell\rangle \in G^I \wedge x_i = y_i \rightarrow \bigwedge_{1 \le j \le \ell} x_j = y_j$$

***Disjointness property*:**

$$\forall x_1, \ldots, x_\ell, y_1, \ldots, y_\ell \in \triangle^I : \langle x_1, \ldots, x_\ell\rangle \in G^I \wedge \langle y_1, \ldots, y_\ell\rangle \in G^I \rightarrow \bigwedge_{1 \le i < j \le n} x_i \neq y_j$$

***A*-start property**: *for each atomic concept* A *with* $V_A \neq \emptyset$,

$$\forall x \in \triangle^I : x \in A^I \rightarrow \exists x_1, \ldots, x_\ell \in \triangle^I : \langle x_1, \ldots, x_\ell\rangle \in G^I \wedge \bigvee_{k \in V_A} x = x_k$$

Vertex layout property: *for each* $i \in V$ *and* $A \in \lambda\langle i\rangle$,

$$\forall x_1, \ldots, x_\ell \in \triangle^I : \langle x_1, \ldots, x_\ell\rangle \in G^I \rightarrow x_i \in A^I$$

Edge layout property: *for each* $\langle i, j\rangle \in E$ *and* $R \in \lambda\langle i, j\rangle$,

$$\forall x_1, \ldots, x_\ell \in \triangle^I : \langle x_1, \ldots, x_\ell\rangle \in G^I \rightarrow \langle x_i, x_j\rangle \in R^I$$

The intuition behind this definition is as follows. Each tuple in the ℓ-ary relation G^I corresponds to one instance of the description graph G.

The i-key properties and the disjointness property ensure that no two instances of G can share an object, which essentially captures the idea behind natural boundedness. Consider the axiom $B \sqsubseteq A$ and a graph G consisting of two vertices 1 and 2 such that $\lambda\langle 1\rangle = \{A\}$, $\lambda\langle 2\rangle = \{B\}$, and $\lambda\langle 1, 2\rangle = \{R\}$. Without the i-key and the disjointness properties, we could build a model I of G where $\gamma_1 \in A^I$,

$\langle \gamma_1, \gamma_2 \rangle \in R^I$, and $\gamma_2 \in B^I$; to ensure that $B^I \subseteq A^I$, we must also set $\gamma_2 \in A^I$; but then, we must instantiate G for γ_2, which clearly leads to a cyclic computation. The i-key and the disjointness properties ensure that such a model I cannot exist: each object occurring in a graph part of I occurs in *exactly* one tuple of G^I. Since this tuple is bounded in size, each graph part of I is bounded as well, which can be used to ensure termination of model construction.

The A-start property ensures that I contains an appropriate instance of G whenever I contains an instance γ of a concept A labeling a vertex of G. If A labels more than one vertex of G, the A-start property "guesses" the vertex of G that γ should be matched to. Consider, for example, a graph containing a vertex labeled with *Hand* and five vertices labeled with *Finger*. If some object γ is an instance of *Finger*, without further information we cannot disambiguate which of the five fingers γ stands for. Therefore, we need to make a "guess" and examine all five possibilities independently. Note that no other property requires guessing; hence, unless we label two vertices in G with the same concept A, the semantic properties of graphs allow for deterministic reasoning.

Finally, the vertex and edge layout properties simply ensure that each instance of G indeed contains the appropriate relational structure of G.

5. REASONING ALGORITHM

To support reasoning over graph-extended KBs, we extend the hypertableau algorithm for $\mathcal{SHIQ}$ from [16]. This algorithm provides the basis for the HermiT reasoner, which is currently the only reasoner that can classify the original version of GALEN. Our algorithm decides satisfiability of $\mathcal{K} = (\mathcal{T}, G, \mathcal{P}, \mathcal{A})$ with $\mathcal{T}$ expressed in $\mathcal{SHIQ}$. The algorithm proceeds in two phases. In the *preprocessing phase*, $\mathcal{T}$ and G are translated into equisatisfiable sets of rules. In the *hypertableau phase*, an attempt is made to construct a model of $\mathcal{A}$, $\mathcal{P}$, and the rules from preprocessing.

The TBox $\mathcal{T}$ is preprocessed into a set of rules $\Xi(\mathcal{T})$ of the form (1) exactly as it is done in [16]. Due to lack of space, we refer the reader for details to [16, 13]. After this transformation, $\exists R.C$ is treated as a synonym for $\geq 1\, R.C$.

Translation of G into a set of rules $\Xi(G)$ uses concepts $\exists G|_k$, which represent objects occurring in an instance of G at position k. Their interpretation is defined as

$$(\exists G|_i)^I = \{s \mid \exists t_1, \ldots, t_\ell : \langle t_1, \ldots, t_\ell \rangle \in G^I \wedge s = t_i\}.$$

The set $\Xi(G)$ is computed as shown in Definition 5. It is easy to see that $\Xi(G)$ encodes conditions from Definition 4 in a straightforward way; hence, $I \models G$ iff $I \models \Xi(G)$.

DEFINITION 5. *For a description graph $G = (V, E, \lambda)$, the set $\Xi(G)$ consists of the following rules, for $\ell = |V|$:*

$$G(x_1, \ldots, x_\ell) \wedge G(y_1, \ldots, y_{i-1}, x_i, y_{i+1}, \ldots, y_\ell) \rightarrow x_j \approx y_j \quad \text{for each } i, j \in V \text{ such that } j \neq i$$

$$G(x_1, \ldots, x_\ell) \wedge G(y_1, \ldots, y_{j-1}, x_i, y_{j+1}, \ldots, y_\ell) \rightarrow \bot \quad \text{for each } 1 \leq i < j \leq \ell$$

$$A(x) \rightarrow \bigvee_{k \in V_A} \exists G|_k(x) \quad \text{for each } A \text{ such that } V_A \neq \emptyset$$

$$G(x_1, \ldots, x_\ell) \rightarrow A(x_i) \quad \text{for each } i \in V \text{ and } A \in \lambda\langle i \rangle$$

$$G(x_1, \ldots, x_\ell) \rightarrow R(x_i, x_j) \quad \text{for } \langle i, j \rangle \in E \text{ and } R \in \lambda\langle i, j \rangle$$

The set of rules $\mathcal{R} = \Xi(\mathcal{T}) \cup \Xi(G) \cup \mathcal{P}$ produced by preprocessing is equisatisfiable with $(\mathcal{T}, G, \mathcal{P})$, so we check satisfiability of $\mathcal{K}$ by checking satisfiability of $(\mathcal{R}, \mathcal{A})$. The hypertableau algorithm, however, can be applied to any set of rules $\mathcal{R}$ that is *admissible* according to Definition 6. It is easy to see that $\Xi(\mathcal{T}) \cup \Xi(G) \cup \mathcal{P}$ is admissible.

DEFINITION 6. *A set of rules $\mathcal{R}$ is* admissible *if it can be represented as a disjoint union of two subsets $\mathcal{R}_t$ and $\mathcal{R}_g$.*

The set $\mathcal{R}_g$ can contain only graph-regular rules and, for each description graph G, it must contain all rules from the first two items of Definition 5.

Let denote R an atomic tree role, A an atomic concept, and C a concept of the form $\geq n\, S.A$ or $\geq n\, S.\neg A$ with S a tree role. For each $r \in \mathcal{R}_t$, it must be possible to separate the variables of r into one center variable x *and the set of* leaf variables $\{y_i\}$ *such that (*i*) each atom in the antecedent of r is of the form $A(x)$, $A(y_i)$, $R(x, y_i)$, or $R(y_i, x)$, (*ii*) each atom in the consequent is of the form $A(x)$, $A(y_i)$, $C(x)$, $C(y_i)$, $R(x, y_i)$, $R(y_i, x)$, or $y_i \approx y_j$, and (*iii*) each variable y_i in the rule occurs in some binary atom in the antecedent.*

Definition 7 summarizes the calculus for checking satisfiability of $(\mathcal{R}, \mathcal{A})$ for $\mathcal{R}$ an admissible set of rules. This algorithm differs from the one in [16] in two main points. First, our algorithm contains the $\exists G$-rule that generates an instance of G for an assertion $\exists G|_i(s)$. In spirit, the $\exists G$-rule is similar to the $\geq$-rule that expands assertions of the form $\geq n\, R.C(s)$ by introducing fresh successors of s. Second, our algorithm contains an appropriately modified version of blocking [10] to ensure termination.

The idea behind blocking is the following: if two individuals s and s' occur in the same concepts in an ABox $\mathcal{A}$, then s "behaves" just like s'—that is, we do not expand s any further. To use this idea in our setting, we separate the individuals in each ABox $\mathcal{A}$ into *named*, *tree*, and *graph* individuals. The named individuals occur in the original graph-extended knowledge base, the tree individuals are introduced by the $\geq$-rule, and the graph individuals are introduced by the $\exists G$-rule. We use this distinction in the definition of blocking: only tree individuals can be blocked, and the blocking individual must also be a tree individual.

Our derivation rules generate only models of the form as shown in Figure 2. There, the individual a is named. The individual h_1 is generated by deriving $\exists \mathit{hasHeart}.\mathit{Heart}(a)$ by (12) and then expanding it by the $\geq$-rule; hence, h_1 is a tree individual. All other individuals that correspond to the structure of the heart (including av) are created by instantiating G, so they are graph individuals. To ensure termination, we apply the $\exists G$-rule with the lowest priority. The rules in $\mathcal{R}$ ensure that no two instances of G can share an individual. Hence, for each tree individual t (such as h_1) that "enters" an instance of G, we can establish a bound on the number of graph individuals occurring generated for t; these individuals are said to be from the same *graph cluster*.

DEFINITION 7. ***Generalized Individuals.*** *Let* T *and* Γ *be two disjoint countably infinite sets of* tree *and* graph symbols. *A* generalized individual *is a finite string of symbols $\alpha_0.\alpha_1.\ \ldots\ .\alpha_n$ such that $\alpha_0 \in N_I$, $\alpha_i \in \mathrm{T} \cup \Gamma$ for $1 \leq i \leq n$, and $\alpha_{i-1} \in \Gamma$ implies $\alpha_i \notin \Gamma$. If $\alpha_n \in N_I$, the individual is* named*; if $\alpha_n \in \mathrm{T}$, the individual is a* tree individual*; and if $\alpha_n \in \Gamma$, the individual is a* graph individual.

Successors and Predecessors. *A generalized individual $x.\alpha$ is a* successor *of x,* predecessor *is the inverse of successor, and* descendant *and* ancestor *are the transitive closures of successor and predecessor, respectively.*

Graph Cluster. *Generalized individuals s and t* are from the same graph clusters *if either (*i*) s is either a named individual or a graph successor of a named individual, and t is also either a named individual or a graph successor of a named individual, (*ii*) both s and t are graph successors of the same tree individual, or (*iii*) one individual is a graph successor of the other individual.*

Generalized ABox. *In the rest of this paper, we allow ABoxes to contain generalized individuals and the assertion $\bot$ which is false in all interpretations, and we take $a \approx b$ ($a \not\approx b$) to also stand for $b \approx a$ ($b \not\approx a$).*

Initial ABox. *An ABox is* initial *if it is extensionally reduced and nonempty and contains only named individuals.*

Pairwise Anywhere Blocking. *A concept is* blocking-relevant *if it is of the form A, $\geq n\, R.A$, $\geq n\, R.\neg A$, or $\exists G|_i$, for A an atomic concept, R a (not necessarily atomic) role, and G a description graph. The* labels *of an individual and of an individual pair in an ABox $\mathcal{A}$ are defined as follows:*

$$\mathcal{L}_{\mathcal{A}}(s) = \{C \mid C(s) \in \mathcal{A} \text{ and } C \text{ is blocking-relevant}\}$$
$$\mathcal{L}_{\mathcal{A}}(s,t) = \{R \mid R(s,t) \in \mathcal{A}\}$$

Let $\prec$ be a a transitive and irreflexive relation on the generalized individuals such that, if s' is an ancestor of s, then $s' \prec s$. By induction on $\prec$, we assign to each individual s in $\mathcal{A}$ a status as follows:

- *s is* directly blocked by *an individual s' iff (*i*) both s and s' are tree individuals, (*ii*) s' is not blocked, (*iii*) $s' \prec s$, (*iv*) $\mathcal{L}_{\mathcal{A}}(s) = \mathcal{L}_{\mathcal{A}}(s')$ and $\mathcal{L}_{\mathcal{A}}(t) = \mathcal{L}_{\mathcal{A}}(t')$, and (*v*) $\mathcal{L}_{\mathcal{A}}(s,t) = \mathcal{L}_{\mathcal{A}}(s',t')$ and $\mathcal{L}_{\mathcal{A}}(t,s) = \mathcal{L}_{\mathcal{A}}(t',s')$, for t and t' the predecessors of s and s', respectively.*
- *s is* indirectly blocked *iff its predecessor is blocked.*
- *s is* blocked *iff it is either directly or indirectly blocked.*

Pruning. *The ABox $\mathsf{prune}_{\mathcal{A}}(s)$ is obtained from $\mathcal{A}$ by removing all assertions that contain a descendant of s.*

Merging. *The ABox $\mathsf{merge}_{\mathcal{A}}(s \to t)$ is obtained from the ABox $\mathsf{prune}_{\mathcal{A}}(s)$ by replacing s with t in all assertions.*

Clash. *An ABox $\mathcal{A}$ contains a* clash *if and only if $\bot \in \mathcal{A}$; otherwise, $\mathcal{A}$ is* clash-free.

Derivation Rules. *Table 1 specifies* derivation rules *that, given a clash-free ABox $\mathcal{A}$ and a set of rules $\mathcal{R}$, derive the ABoxes $\langle \mathcal{A}_1, \ldots, \mathcal{A}_n \rangle$. In the Hyp-rule, σ is a mapping from the set of variables N_V to the individuals in $\mathcal{A}$, and $\sigma(U)$ is obtained from U by replacing each variable x with $\sigma(x)$.*

Rule Priority. *The $\exists G$-rule is applicable only if no other rule is applicable.*

Derivation. *A* derivation *for a set of admissible rules $\mathcal{R}$ and an initial ABox $\mathcal{A}$ is a pair (T, ρ) where T is a finitely branching tree and ρ labels the nodes of T with ABoxes such that (*i*) $\rho(\epsilon) = \mathcal{A}$ for ϵ the root of the tree, and (*ii*) for each node t, if one or more derivation rules are applicable to $\rho(t)$ and $\mathcal{R}$, then t has children $t_1, \ldots, t_n$ such that the ABoxes $\langle \rho(t_1), \ldots, \rho(t_n) \rangle$ are the result of applying one applicable derivation rule chosen by respecting the rule priority. A derivation is* clash-free *if it has a leaf node labeled with a clash-free ABox.*

THEOREM 1. *For an admissible set of rules $\mathcal{R}$ and an initial ABox $\mathcal{A}$, (*i*) if $(\mathcal{R}, \mathcal{A})$ is satisfiable, then each derivation for $\mathcal{R}$ and $\mathcal{A}$ is clash-free, (*ii*) if a clash-free derivation for $\mathcal{R}$ and $\mathcal{A}$ exists, then $(\mathcal{R}, \mathcal{A})$ is satisfiable, and (*iii*) each derivation for $\mathcal{R}$ and $\mathcal{A}$ is finite.*

The proof is given in the technical report [13]. As a consequence of the theorem, subsumption checking, concept satisfiability, and query answering are decidable as well.

6. FROM OWL AXIOMS TO GRAPHS

The evaluation of the adequacy of our approach is rather difficult due to lack of adequate test data. Furthermore, remodeling existing ontologies using a new modeling paradigm may require considerable effort. In order to both obtain test data for our reasoner and make the adoption of our approach in practice easier, we have developed an algorithm that automatically transforms a TBox $\mathcal{T}$ into a graph-extended knowledge base $\mathcal{K}$. For example, our algorithm can automatically construct the graph shown in Figure 1(a) from the axioms such as (2)–(4). Clearly, the knowledge base $\mathcal{K}$ is only a rough approximation; however, it can be used as a starting point for a more comprehensive remodeling of $\mathcal{T}$ into a proper graph-extended KB.

6.1 The Transformation Algorithm

Our transformation of a TBox $\mathcal{T}_1$ into a graph-extended KB $\mathcal{K} = (\mathcal{T}, G, \mathcal{P}, \mathcal{A})$ is based on two assumptions.

The first assumption is that only some concepts and roles from $\mathcal{T}_1$ are *relevant* for G. For example, the *Heart* concept is clearly relevant to the description graph of human anatomy; in contrast, the *Disease* concept is not relevant because it does not represent the structure of a human body. Similarly, the *hasStructuralComponent* role clearly belongs to the graph, while the *hasAge* role does not.

Our second assumption is that each concept relevant to G should be represented by one vertex in G, and that edges in G can be decoded from axioms of the form $A \sqsubseteq \exists R.B$. Our assumption is that, by writing axioms such as (2)–(4), modelers actually wanted to say "the aortic valve has an alpha connection to the left ventricle, and the left side of heart has *the same* left ventricle as a solid division."

We use these two assumptions in the core part of our algorithm, which is parameterized with a DL TBox $\mathcal{T}_1$, a set of relevant concepts N_{C_g}, and a set of relevant roles N_{R_g}. The latter set actually defines the set of graph roles, and all other roles are considered to be tree roles. Our algorithm first normalizes $\mathcal{T}_1$ in a certain way. Then, it creates a vertex i in V for each concept $A \in N_{C_g}$ and sets $\lambda\langle i \rangle = \{A\}$. Then, it processes each axiom $\alpha \in \mathcal{T}_1$ as follows:

- If α is of the form $A \sqsubseteq \exists R.B$ where $\{A, B\} \subseteq N_{C_g}$ and $R \in N_{R_g}$, then, for i and j vertices such that $\lambda\langle i \rangle = \{A\}$ and $\lambda\langle j \rangle = \{B\}$, the algorithm adds the edge $\langle i, j \rangle$ to E and extends λ such that $R \in \lambda\langle i, j \rangle$.
- If α does not contain a role from N_{R_g}, the algorithm simply copies α to the resulting TBox $\mathcal{T}$.
- If α contains only roles from N_{R_g} and no existential quantifier, the algorithm translates α into a graph-regular rule and adds it to $\mathcal{P}$.
- If α is not of the above form, then either it involves a graph and a tree role simultaneously, or it is of the form $A \sqsubseteq \exists R.B$ but some of A, B, or R are not relevant for the graph. Such an axiom either invalidates the syntactic restrictions of our formalism or it does not have a natural interpretation; hence, our algorithm simply ignores such an axiom α.

Table 1: Derivation Rules of the Hypertableau Calculus

Hyp-**rule**

If 1. $U_1 \wedge ... \wedge U_m \rightarrow V_1 \vee ... \vee V_n \in \mathcal{R}$,
2. a mapping $\sigma : N_V \rightarrow N_\mathcal{A}$ exists such that
2.1 $\sigma(U_i) \in \mathcal{A}$ for each $1 \leq i \leq m$ and
2.2 $\sigma(V_j) \notin \mathcal{A}$ for each $1 \leq j \leq n$,
then $\mathcal{A}_1 = \mathcal{A} \cup \{\bot\}$ if $n = 0$; or
$\mathcal{A}_j := \mathcal{A} \cup \{\sigma(V_j)\}$ for $1 \leq j \leq n$ if $n > 0$.

$\approx$**-rule**

If $s \approx t \in \mathcal{A}$ and $s \neq t$
then $\mathcal{A}_1 := \mathsf{merge}_\mathcal{A}(s \rightarrow t)$ if t is a named individual or if s is a descendant of t; or
$\mathcal{A}_1 := \mathsf{merge}_\mathcal{A}(t \rightarrow s)$ otherwise.

$\bot$**-rule**

If $s \not\approx s \in \mathcal{A}$ or $\{A(s), \neg A(s)\} \subseteq \mathcal{A}$
then $\mathcal{A}_1 := \mathcal{A} \cup \{\bot\}$.

$\geq$**-rule**

If 1. $\geq n\, R.C(s) \in \mathcal{A}$,
2. s is not blocked in $\mathcal{A}$, and
3. there are no individuals $u_1, \ldots, u_n$ such that $\{\mathsf{ar}(R, s, u_i), C(u_i) \mid 1 \leq i \leq n\} \cup \{u_i \not\approx u_j \mid 1 \leq i < j \leq n\} \subseteq \mathcal{A}$,
then $\mathcal{A}_1 := \mathcal{A} \cup \{\mathsf{ar}(R, s, t_i),\ C(t_i) \mid 1 \leq i \leq n\} \cup \{t_i \not\approx t_j \mid 1 \leq i < j \leq n\}$ where $t_1, \ldots, t_n$ are fresh pairwise distinct tree successors of s.

$\exists G$**-rule**

If 1. $\exists G|_i(s) \in \mathcal{A}$ for G a description graph with ℓ vertices,
2. s is not blocked in $\mathcal{A}$, and
3. there are no individuals $u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_\ell$ such that $G(u_1, \ldots, u_{i-1}, s, u_{i+1}, \ldots, u_\ell) \in \mathcal{A}$
then $\mathcal{A}_1 := \mathcal{A} \cup \{G(t_1, \ldots, t_{i-1}, s, t_{i+1}, \ldots, t_\ell)\}$ where $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_\ell$ are fresh pairwise distinct graph individuals from the same graph cluster as s.

Note: $\mathcal{A}$ is a generalized ABox, $\mathcal{R}$ is a set of admissible rules, and $N_\mathcal{A}$ is the set of individuals occurring in $\mathcal{A}$.

Our translation cannot correctly handle axioms of the form $A \sqsubseteq \geq n\, R.B$ with $n \geq 2$. Intuitively, such axioms might be handled by creating n vertices in G, labeling all of them with B, and then connecting the vertex of A with all the vertices of B using the role R. The situation, however, is not so simple if, in addition, we also have the axiom $B \sqsubseteq \geq m\, R.A$. It is now not clear which vertices of the description graph labeled with A to "reuse" to satisfy this axiom. Therefore, we decided to ignore such axioms. This is partly justified by the fact that GALEN and FMA—our main sources of inspiration and test data—do not contain $\geq n\, R.B$ concepts with $n \geq 2$. In human anatomy, different objects of the domain are naturally given different names. For example, instead of an axiom

$$\mathit{Heart} \sqsubseteq \geq 2\, \mathit{hasStructuralComponent}.\mathit{SideOfHeart}, \quad (16)$$

GALEN introduces explicit names for the left and the right side of the heart:

$$\mathit{Heart} \sqsubseteq \exists \mathit{hasStructuralComponent}.\mathit{LeftSideOfHeart} \quad (17)$$
$$\mathit{Heart} \sqsubseteq \exists \mathit{hasStructuralComponent}.\mathit{RightSideOfHeart} \quad (18)$$

On ontologies with at-least restrictions, our algorithm simply treats each $\geq n\, R.B$ as $\exists R.B$. It is natural to use number restrictions for modeling symmetric organs such as the kidney. On such an ontology, our algorithm produces a description graph containing just one copy of the object, and the graph can then be corrected by the modeler.

Determining the sets N_{C_g} and N_{R_g} manually is not easy. According to our experience with GALEN and FMA, a good strategy is to manually identify a set of roles N'_{R_g} that naturally belong to the graph, and then to take N_{R_g} as the closure of N'_{R_g} w.r.t. the explicit role inclusions from $\mathcal{T}_1$. Then, we take N_{C_g} as the set of all concepts A and B occurring in an axiom $A \sqsubseteq \exists R.B \in \mathcal{T}_1$ such that $R \in N_{R_g}$. Intuitively, if A and B are connected by a role that should be included into the graph, then it is likely that A and B should be included into the graph as well.

This idea, however, requires some refinement. For example, GALEN contains the following axioms:

$$\mathit{LeftVentricle} \sqsubseteq \mathit{Ventricle} \quad (19)$$
$$\mathit{RightVentricle} \sqsubseteq \mathit{Ventricle} \quad (20)$$

Let us assume that N_{C_g} contains *Ventricle*, *LeftVentricle*, and *RightVentricle*. The core transformation then generates a description graph G containing three different vertices, each labeled with one of these concepts. It is, however, counterintuitive for G to contain a *Ventricle* vertex: no ventricle as such exists on its own; rather, each concrete ventricle is either the left of the right ventricle. In fact, such a description graph G is unsatisfiable. Assume that an object x as instance of *LeftVentricle*; due to (19), x is also an instance of *Ventricle*. To satisfy the A-start property for *LeftVentricle*, x must correspond to the i-th vertex of some instance of G; similarly, to satisfy the A-start property for *Ventricle*, x must also correspond to the j-th vertex of some instance of G. Finally, because *LeftVentricle* and *Ventricle* label different vertices of G, we have $i \neq j$, which then invalidates the disjointness property of Definition 4. The concept *Ventricle* is thus an *abstract concept*: it is not meant to be instantiated directly, but only through a subclass. Such concepts clearly do not belong into a description graph. Hence, after computing N_{C_g} as described in the previous paragraph, our algorithm classifies the input TBox $\mathcal{T}_1$ using standard DL reasoning; then, it removes from N_{C_g} all concepts that are not leaves in the resulting classification. Intuitively, if A is not a leaf concept in the classification of $\mathcal{T}_1$, then A is likely to be an abstract concept, so it should not be added to G.

A pseudo-code of the algorithm is given in [13], and the tool can be downloaded from HermiT's Web site.

6.2 Transforming GALEN and FMA

We applied the algorithm from Section 6.1 to the original version of GALEN; furthermore, FMA is a very large ontology, so we have applied our algorithm to a fragment of FMA that describes the heart. Both ontologies can be downloaded from HermiT's Web page. Table 2 summarizes information about the original and the transformed ontologies.

Our transformation clearly leads to a change in the semantics of the ontology, and some information is lost in the process. Many parts of the resulting description graph, however, correspond with the intuitive descriptions of the anatomy of the body. For example, the graph shown in Figure 1(a) has been extracted from the transformed ontology.

7. EVALUATION AND DISCUSSION

To evaluate our approach, we have classified the original ontologies using HermiT, transformed them using the algorithm from Section 6 into graph-extended KBs, and classified the resulting KBs using the reasoning algorithm pre-

Table 2: Information about Test Ontologies

	GALEN	FMA
Total number of concepts:	2748	430
Total number of roles:	413	38
Total number of GCIs:	6962	3479
GCIs discarded in the transformation:	320	328
With both a tree and a graph role:	74	0
With existentials on abstract concepts:	246	328
Translated GCIs:	6642	3151
Into the description graph:	680	2966
Into rules over the graph:	155	1
Into the DL TBox:	5807	184
With existentials on tree roles:	1741	16
With universals on tree roles:	952	0
Involving concept names only:	3114	168
Vertices in the description graph:	325	342
Edges in the description graph:	667	1076

sented in Section 5. We now present the performance results and discuss the classification results.

7.1 Performance Results

We performed the experiments using a standard laptop with 1 GB of RAM. The classification of the original version of GALEN and the fragment of FMA took 129 s and 57 s, respectively; furthermore, the classification of the transformed ontologies took 781 s and 6 s, respectively.

The increase in the classification time for GALEN is partly due to the fact that our implementation of the reasoning algorithm in Section 5 is still very prototypical. In the case of FMA, the classification times are substantially lower because most of the original ontology is translated into the graph, so the generated models are much smaller.

Our performance results show that, even with a very prototypical implementation, we can process complex ontologies, which we take as indication that our approach is practically feasible.

7.2 Changes in the Semantics

The transformed ontologies are more constrained than the original ones, so we expect to obtain new entailments.

In the case of GALEN, we discovered a concept that is satisfiable in the original version of the ontology, but is unsatisfiable in the transformed ontology, which revealed a modeling error in GALEN. The problem occurs in the representation of the patella—a bone that is connected to certain tendons through two retinacula, represented using the concepts *LateralPatellaRetinaculum* and *MedialPatellaRetinaculum*. GALEN describes the relationship between the patella and the two retinacula as follows:

$$\mathit{LateralPatellaRetinaculum} \equiv \exists \mathit{hasOtherEndAt}.\mathit{Patella} \sqcap \ (\ldots) \tag{21}$$

$$\mathit{MedialPatellaRetinaculum} \equiv \exists \mathit{hasOtherEndAt}.\mathit{Patella} \sqcap \ (\ldots) \tag{22}$$

$$\mathit{hasOtherEndAt} \equiv \mathit{isAtOtherEndOf}^{-} \tag{23}$$

$$\top \sqsubseteq \leq 1\, \mathit{isAtOtherEndOf} \tag{24}$$

According to the axioms above, each patella is connected to both the lateral and the medial retinacula, but due to functionality of *isAtOtherEndOf*, the two must be the same objects. Intuitively, this is an undesirable consequence, since the two retunaculae are in reality different objects; in other words, *isAtOtherEndOf* should probably not have been declared functional. Since GALEN is underconstrained, this does not cause the inconsistency of either concept, so this error has not been detected so far. The description graph produced by our transformation, however, contains one node for the patella and one for each retinaculum; furthermore, both retinacula are connected through *isAtOtherEndOf* to the same patella. Since *isAtOtherEndOf* is functional, the retinacula should be the same, which invalidates the disjointness property for the graph (see Definition 4) and makes *Patella* unsatisfiable.

In the case of FMA, we did not obtain any new subsumption relationships. This is due to the fact that most of the subsumption relationships in FMA are represented explicitly as axioms of the form $A \sqsubseteq B$ where A and B are atomic concepts. For example, the fact that the heart is an organ is represented explicitly with the axiom $\mathit{Heart} \sqsubseteq \mathit{Organ}$, and it is not derived from the structure of the heart; clearly, such inferences are performed in the same way on both tree-like and nontree structures.

As explained in Section 6, our algorithm discards some axioms from the ontology. We compared the class hierarchies of the original and the graph-extended versions of GALEN. In total, 361 subsumption relationships were lost, such as $\mathit{Femur} \sqsubseteq \mathit{BodySpace}$ (the femur is a body space), and

$$\mathit{InteratrialSeptum} \sqsubseteq \mathit{TwoAndAHalfDimensionalStructure}$$

(the interatrial septum of the heart is a structure with two and a half dimensions). All these entailments involve an abstract concept, so their loss is not surprising since the transformation algorithm discards GCIs that involve an abstract concept and an existential on a graph role. No information about concrete concepts has been lost, though.

In contrast, in the case of FMA we did not lose any subsumption relationships. As explained before, the reason is that the structural information in FMA largely does not influence subsumption.

7.3 Discussion

Our experience with GALEN and the discussions we had with the authors of GALEN lead us to conclude that our formalism represents the anatomical structures in the human body in a way that is closer to the modelers' intention than the original OWL axioms.[3] The fact that we found a modeling error in GALEN leads us to believe that our formalism and its semantics are based on "reasonable" assumptions.

Furthermore, capturing the semantics of abstract concepts and axioms involving them properly is likely to be the most important open problem. We briefly discuss possibilities for addressing it. Consider the following axiom in GALEN that is eliminated by the transformation algorithm because both *AtrioventricularValve* and *Ventricle* are abstract concepts:

$$\mathit{AtrioventricularValve} \sqsubseteq \exists \mathit{hasAlphaConnection}.\mathit{Ventricle} \tag{25}$$

Since both concepts in (25) are abstract, this axiom does not say anything about the structure of the concrete objects (i.e., the objects that are likely to be included into a description graph). Thus, one might expect the actual relationship between valves and ventricles to be described for the concrete subclasses of *AtrioventricularValve* and *Ventricle*. Axiom (25) can then be interpreted as a check which makes

[3]Thanks to Alan Rector and Sebastian Brandt.

sure that this abstract relationship is concretized at a lower level. Another possibility is to interpret *Ventricle* disjunctively over its subclasses: each valve is connected to either left or the right ventricle, but we do not know which. Currently, it is not clear which interpretation is appropriate; in fact, the proper interpretation of abstract concepts is made more difficult by the fact that whether a concept is abstract or concrete depends on the level of granularity.

8. CONCLUSION

We have extended OWL with description graphs, which can be used to describe structured objects—objects consisting of parts connected in a complex, arbitrary way. We also allow for arbitrary SWRL-like rules over description graphs. Unlike most existing combinations of DLs and rules in which rules can be used only for query answering [12, 15, 17, 6, 14], our rules also fully participate in schema reasoning. Based on an observation that many structured objects exhibit a natural bound on their size, we derived a hypertableau reasoning algorithm for our formalism, which we implemented in the HermiT reasoner. To obtain suitable test data, we extracted description graphs out of GALEN and FMA medical terminologies. We successfully classified the resulting ontologies and even detected a modeling error.

We see three open problems for future research. First, graph-extended KBs should provide for several and not just one description graph, as this would allow breaking up a large graph into several more manageable parts. The main challenge is to identify an appropriate paradigm for specifying relationships between different description graphs. Second, an adequate semantics for modeling abstract concepts at different levels of granularity is needed. Third, to allow for a wider users' community, we would need to extend ontology editors such as Protégé with description graphs.

Acknowledgments

We thank Alan Rector and Sebastian Brandt for providing us with comments from the domain experts' perspective.

9. REFERENCES

[1] A. Artale, E. Franconi, N. Guarino, and L. Pazzi. Part-whole relations in object-centered systems: An overview. *Data Knowledge & Engineering*, 20(3):347–383, 1996.

[2] F. Baader, C. Lutz, H. Sturm, and F. Wolter. Fusions of Description Logics and Abstract Description Systems. *Journal of Artificial Intelligence Research*, 16:1–58, 2002.

[3] D. Calvanese, G. De Giacomo, and M. Lenzerini. Structured Objects: Modeling and Reasoning. In *Proc. DOOD '95*, pages 229–246, 1995.

[4] S. Derriere, A. Richard, and A. Preite-Martinez. An Ontology of Astronomical Object Types for the Virtual Observatory. In *Proc. of the 26th meeting of the IAU*, pages 17–18, 2006.

[5] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. AL-log: Integrating Datalog and Description Logics. *Journal of Intelligent Information Systems*, 10(3):227–252, 1998.

[6] T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining Answer Set Programming with Description Logics for the Semantic Web. In *Proc. KR 2004*, pages 141–151, 2004.

[7] F. W. Hartel, S. de Coronado, R. Dionne, G. Fragoso, and J. Golbeck. Modeling a description logic vocabulary for cancer research. *Journal of Biomedical Informatics*, 38(2):114–129, 2005.

[8] I. Horrocks and P. F. Patel-Schneider. A Proposal for an OWL Rules Language. In *Proc. WWW 2004*, pages 723–731, 2004.

[9] I. Horrocks and U. Sattler. A Tableaux Decision Procedure for $\mathcal{SHOIQ}$. In *Proc. IJCAI 2005*, pages 448–453, 2005.

[10] I. Horrocks, U. Sattler, and S. Tobies. Practical Reasoning for Very Expressive Description Logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.

[11] O. Kutz, I. Horrocks, and U. Sattler. The Even More Irresistible SROIQ. In *Proc. KR 2006*, pages 68–78, 2006.

[12] A. Y. Levy and M.-C. Rousset. Combining Horn Rules and Description Logics in CARIN. *Artificial Intelligence*, 104(1–2):165–209, 1998.

[13] B. Motik, B. Cuenca Grau, and U. Sattler. Representation of and Reasoning with Structured Objects in OWL. Technical report, University of Oxford, UK, 2007. See first author's Web page.

[14] B. Motik and R. Rosati. A Faithful Integration of Description Logics with Logic Programming. In *Proc. IJCAI 2007*, pages 477–482, 2007.

[15] B. Motik, U. Sattler, and R. Studer. Query Answering for OWL-DL with Rules. *Journal of Web Semantics*, 3(1):41–60, 2005.

[16] B. Motik, R. Shearer, and I. Horrocks. Optimized Reasoning in Description Logics using Hypertableaux. In *Proc. CADE-21*, pages 67–83, 2007.

[17] R. Rosati. $\mathcal{DL} + log$: A Tight Integration of Description Logics and Disjunctive Datalog. In *Proc. KR 2006*, pages 68–78, 2006.

[18] C. Rosse and J. V. L. Mejino. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *Journal of Biomedical Informatics*, 36:478–500, 2003.

[19] J. Seidenberg and A. L. Rector. Representing Transitive Propagation in OWL. In *Proc. ER 2006*, pages 255–266, 2006.

[20] A. Sidhu, T. Dillon, E. Chang, and B. Singh Sidhu. Protein Ontology Development using OWL. In *Proc. OWLED 2005*, 2005.

[21] D. Soergel, B. Lauser, A. Liang, F. Fisseha, J. Keizer, and S. Katz. Reengineering Thesauri for New Applications: The AGROVOC Example. *Journal of Digital Information*, 4(4), 2004.

[22] W.D. Solomon, A. Roberts, J. E. Rogers, C. J. Wroe C.J., and A. L. Rector. Having our cake and eating it too: How the GALEN Intermediate Representation reconciles .... In *Proc. AMIA*, pages 819–823, 2000.

[23] K. A Spackman. SNOMED RT and SNOMEDCT. Promise of an international clinical terminology. *M.D. Computing*, 17(6):29, 2000.

[24] M. Y. Vardi. Why Is Modal Logic So Robustly Decidable? In *Proc. DIMACS Workshop*, volume 31, pages 149–184, 1996.