The Representation of Structured Objects in DLs using Description Graphs*

Boris Motik[1], Bernardo Cuenca Grau[1], and Ulrike Sattler[2]

[1] Computing Laboratory, University of Oxford, UK
[2] School of Computer Science, University of Manchester, UK

1 Introduction

Applications of description logics (DLs) often require the representation of and reasoning with *structured objects*—that is, objects composed of parts connected in complex ways. Although DLs are general and powerful languages, they cannot describe arbitrarily connected structures. The description of structured objects in DLs can thus be underconstrained, which reduces the number of entailments and can even cause performance problems for reasoning. Hence, we propose an extension of DLs with *description graphs*, which allow structured objects to be described in a simple and precise way. To represent conditional aspects of the domain, we also allow for Horn rules over description graphs.

Extending DLs with axioms that can enforce arbitrary structures easily leads to undecidability [5]. Our formalism, however, is decidable because it can represent only structured objects whose number of parts is bounded. In practice, structured objects are usually modeled up to a certain level of granularity, which naturally determines this bound. We present a reasoning algorithm for the case where the DL part is expressed in $\mathcal{SHIQ}$ [3]. We thus obtain a powerful, decidable, and practicable language that combines two complementary formalisms: unbounded but tree-like structures can be described using standard DL axioms, and the naturally bounded structured parts can be described using arbitrarily connected description graphs and rules. Due to lack of space, we cannot present the correctness proofs in this paper; we refer the interested reader to [6].

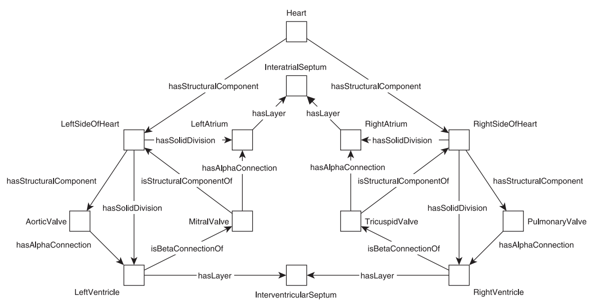
We have implemented our procedure in the HermiT reasoner.[3] Furthermore, we have extracted description graphs from the GALEN and FMA ontologies, classified them successfully, and even detected a modeling error in GALEN.

2 Motivation

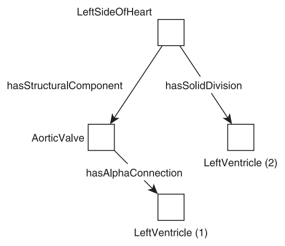
Structured objects—objects composed of parts connected in complex ways—abound in molecular biology and the clinical sciences. Clinical ontologies such as GALEN [10] and the Foundation Model of Anatomy (FMA) describe numerous

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

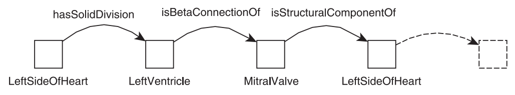
[3] `http://web.comlab.ox.ac.uk/oucl/work/boris.motik/HermiT/`



(a) Intended Structure



(b) Unintended Structure



(c) Unintended Infinite Structure

Fig. 1. Different Models of the Heart in GALEN

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.” Figure 1(a) represents the intention behind the axioms describing the heart in GALEN.

This figure could be represented in DLs 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 consider modeling structured objects *at the TBox level*—that is, we want to describe the general structure of *all* hearts and then instantiate this description as often as needed. This clearly cannot be achieved if we describe the structure of the heart using ABox assertions.

We can give a logical, TBox-level interpretation to Figure 1(a) in DLs by treating vertices as concepts and arrows as *participation constraints*. 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 encoded in DLs using axioms of the form (1). Let $\mathcal{K}$ be a DL KB containing the axioms (1)–(3), and let I be an interpretation that corresponds to Figure 1(a) in the obvious way. Clearly, $I \models \mathcal{K}$, which justifies the formalization of Figure 1(a) by $\mathcal{K}$.

$$\mathit{LeftSideOfHeart} \sqsubseteq \exists \mathit{hasStructuralComponent}.\mathit{AorticValve} \tag{1}$$

$$\mathit{AorticValve} \sqsubseteq \exists \mathit{hasAlphaConnection}.\mathit{LeftVentricle} \tag{2}$$

$$\mathit{LeftSideOfHeart} \sqsubseteq \exists \mathit{hasSolidDivsion}.\mathit{LeftVentricle} \tag{3}$$

We can now describe various heart conditions; for example, if the aortic valve suffers from aortic regurgitation (AR), then the left ventricle suffers from left ventricular hypertrophy (LVH), cf. (4). Then, from (1)–(4) we might want to derive that, if the aortic valve of the left side of the heart has aortic regurgitation, then the left ventricle suffers from hypertrophy, cf. (5).

$$\mathit{AorticValve} \sqcap \mathit{HasAR} \sqsubseteq \forall \mathit{hasAlphaConnection}.\mathit{HasLVH} \tag{4}$$

$$\begin{aligned}\mathit{LeftSideOfHeart} \sqcap \exists \mathit{hasStructuralComponent}.(\mathit{AorticValve} \sqcap \mathit{HasAR})\\ \sqsubseteq \exists \mathit{hasSolidDivision}.\mathit{HasLVH}\end{aligned} \tag{5}$$

Unfortunately, (5) does not follow from $\mathcal{K}$: axioms (2) and (3) 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, $I' \not\models (5)$, so $\mathcal{K} \not\models (5)$ 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 (5). Furthermore, it can also cause problems with the performance of reasoning. For example, we might use axioms (3) and (6)–(7) to describe the relationships between the left side of the heart, the left ventricle, and the mitral valve.

$$\mathit{LeftVentricle} \sqsubseteq \exists \mathit{isBetaConnectionOf}.\mathit{MitralValve} \tag{6}$$

$$\mathit{MitralValve} \sqsubseteq \exists \mathit{isStructuralComponentOf}.\mathit{LeftSideOfHeart} \tag{7}$$

These axioms do not state that the mitral valve in (6) is a structural component of the "initial" left side of the heart. Thus, the interpretation from Figure 1(c) is a "canonical" model of these axioms—it is "canonical" in the sense that it reflects the least amount of information derivable from the axioms. In order to disprove an entailment, a DL reasoner will try to construct such a model, which can be much larger than the intended one. According to our experience, this is a major source of performance problems with reasoning.

Therefore, we need to extend $\mathcal{K}$ with additional axioms that make *all* models of $\mathcal{K}$ as close as possible to the intended conceptualization in Figure 1(a). This, however, is not possible in DLs. DLs can represent *unbounded* or even *infinite* domains. For example, 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." To ensure termination of reasoning with such axioms, DLs typically exhibit (a variant of) the *tree model property*. The relationship between the left side of the heart, the aortic valve, and the left ventricle, however, is triangular. Hence, to ensure that the ventricles implied by (2) and (3) are the same in *every* model of $\mathcal{K}$, we must leave the confines of traditional DLs. $\mathcal{SROIQ}$ addresses this problem to a certain extent [4], but it still does not allow us to axiomatize arbitrarily connected structures.

Rule formalisms can axiomatize nontree structures. For example, the following SWRL [2] rule forces the two ventricles from Figure 1(b) to be the same:

$$\begin{aligned}&\mathit{LeftSideOfHeart}(x) \wedge \mathit{hasStructuralComponent}(x,y) \wedge\\ &\mathit{hasAlphaConnection}(y,z) \wedge \mathit{LeftVentricle}(z) \wedge\\ &\mathit{hasSolidDivsion}(x,w) \wedge \mathit{LeftVentricle}(w) \rightarrow z \approx w\end{aligned} \tag{8}$$

From the standpoint of modeling, the equality of the two left ventricles is in (8) not represented explicitly: it follows from the complex interaction between axioms (1)–(3) and (8). This modeling style is hard to use in practice and susceptible to modeling errors. From the standpoint of automated reasoning, SWRL is undecidable [2], which is a significant impediment to its adoption 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. Such a rule cannot be expressed in DLs since it implies non-tree-like antecedents. If we, however, deal with arbitrarily connected structures, such as the one shown in Figure 1(a), such rules are essential for drawing the correct inferences.

Various decidable combinations of DLs and rules cannot be used for TBox modeling. For example, the DL-safe rules [8] are syntactically restricted such that they apply only to the explicitly named objects. Role-safe [5] and weakly safe [9] rules also impose restrictions that prevent the application of the rules to arbitrary elements of the domain. While these can be useful in answering ABoxes queries, they have no effect on TBox entailments.

3 Description Graphs

We propose a formalism that can overcome some of the problems identified in Section 2. The main idea is that parts of an object can naturally be represented by a graph. All subsequent definitions are parameterized by a *graph-extended DL signature*, which consists 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.

Definition 1. *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* $i \in V$ *with a set* $\lambda\langle i\rangle \subseteq N_C$*, and each* $\langle i, j\rangle \in E$ *with a set* $\lambda\langle i, j\rangle \subseteq N_{R_g}$*. 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 2. *A rule is an expression of the form* (9) *where* B_i *and* H_j *are atoms. A rule is* graph-regular *if it uses only atomic concepts and graph roles and if each pair of variables from the rule occurs in some* B_i*.*

$$B_1 \wedge \ldots \wedge B_n \rightarrow H_1 \vee \ldots \vee H_m \tag{9}$$

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 ABox over* $(N_C, N_{R_t}, N_{R_g}, N_I)$ *that can also contain* graph assertions *of the form* $G(a_1, \ldots, a_\ell)$ *for* $a_i \in N_I$*.*

For example, let $\mathcal{K} = (\mathcal{T}, G, \mathcal{P}, \mathcal{A})$ be a graph-extended DL knowledge base where $\mathcal{T}$ contains the axioms (10)–(12). Intuitively, axiom (10) says that each person has a parent and a heart; axiom (11) ensures that the heart of each sufferer from aortic regurgitation is an instance of *HasAR*; and axiom (12) says that, on each aortic valve suffering from aortic regurgitation, some person is performing a surgery on it.

$$Person \sqsubseteq \exists hasParent.Person \sqcap \exists hasHeart.Heart \tag{10}$$

$$AR\_Sufferer \sqsubseteq \forall hasHeart.HasAR \tag{11}$$

$$AorticValve \sqcap HasAR \sqsubseteq \exists performsSurgeryOn^{-}.Person \tag{12}$$

Let G correspond to Figure 1(a), let $\mathcal{P}$ contain the rule (13) that propagates the *HasAR* concept over the structural components of the heart, and let $\mathcal{A}$ contain the assertions $Person(a)$ and $AR\_Sufferer(a)$.

$$HasAR(x) \wedge hasStructuralComponent(x, y) \rightarrow HasAR(y) \tag{13}$$

We now define the semantics of graph-extended KBs.

Definition 3. *An interpretation $I = (\triangle^I, \cdot^I)$ interprets a 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 $1 \leq i \leq \ell$, and $\forall x_1, \ldots, x_\ell, y_1, \ldots, y_\ell \in \triangle^I$:*
$\langle x_1, \ldots, x_\ell \rangle \in G^I \wedge \langle y_1, \ldots, y_\ell \rangle \in G^I \wedge x_i = y_i \rightarrow \bigwedge_{1 \leq j \leq \ell} x_j = y_j$

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

A-start property*: for each atomic conceptA 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 $R \in \lambda\langle i, j \rangle$,*
$\forall x_1, \ldots, x_\ell \in \triangle^I : \langle x_1, \ldots, x_\ell \rangle \in G^I \rightarrow \langle x_i, x_j \rangle \in R^I$

Each model I of our example knowledge base $\mathcal{K}$ consists of two distinct parts, as shown in Figure 2. 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 $\mathcal{T}$. 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 in the case of axioms (1)–(3) 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

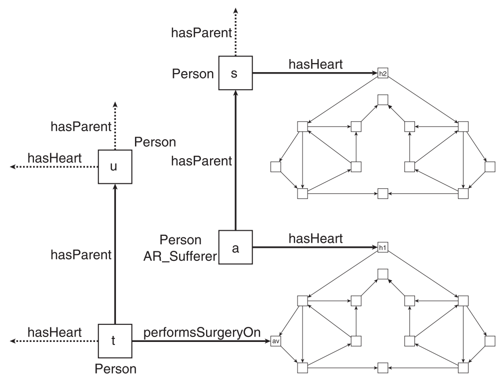


Fig. 2. A Typical Model of $\mathcal{K}$

contiguous: the bottom-most *AorticValve* object *av* can be connected to other objects through tree roles.

Each tuple in the ℓ-ary relation G^I corresponds to one instance of the description graph G. The i-key and the disjointness properties ensure that each object occurring in a graph part of I occurs in *exactly* one tuple of G^I, which essentially captures the idea behind natural boundedness. 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. The vertex and edge layout properties simply ensure that each instance of G indeed contains the appropriate relational structure of G.

4 Reasoning Algorithm

To support reasoning over graph-extended KBs, we derived an algorithm that 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 obtained in preprocessing.

The TBox $\mathcal{T}$ is preprocessed into a set of rules $\Xi(\mathcal{T})$ of the form (9). The translation of G into a set of rules $\Xi(G)$ uses concepts $\exists G|_k$, which are interpreted 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 4, and it captures the conditions from Definition 3 in a straightforward way; hence, $I \models G$ iff $I \models \Xi(G)$.

Definition 4. *For a description graph $G = (V, E, \lambda)$, the set $\Xi(G)$ consists of the following rules, for $\ell = |V|$:*

$$
\begin{array}{ll}
G(x_1, \ldots, x_\ell) \wedge G(y_1, \ldots, y_{i-1}, x_i, y_{i+1}, \ldots, y_\ell) \rightarrow x_j \approx y_j & \textit{for each } i, j \in V \\
G(x_1, \ldots, x_\ell) \wedge G(y_1, \ldots, y_{j-1}, x_i, y_{j+1}, \ldots, y_\ell) \rightarrow \bot & \textit{for each } 1 \leq i < j \leq \ell \\
A(x) \rightarrow \bigvee_{k \in V_A} \exists G|_k(x) & \textit{for each } A \textit{ such that } V_A \neq \emptyset \\
G(x_1, \ldots, x_\ell) \rightarrow A(x_i) & \textit{for each } i \in V \textit{ and } A \in \lambda\langle i \rangle \\
G(x_1, \ldots, x_\ell) \rightarrow R(x_i, x_j) & \textit{for } \langle i, j \rangle \in E \textit{ and } R \in \lambda\langle i, j \rangle
\end{array}
$$

The hypertableau algorithm can be applied to any set of rules $\mathcal{R}$ that are *admissible* according to a certain criteria [6]. Intuitively, an admissible set of rules $\mathcal{R}$ must consist of a subset $\mathcal{R}_g$ of graph-regular rules and of a subset of $\mathcal{R}_t$ of *tree* rules that are obtained from the DL axioms in a TBox $\mathcal{T}$. The set of rules $\mathcal{R} = \Xi(\mathcal{T}) \cup \Xi(G) \cup \mathcal{P}$ is admissible.

Definition 5 summarizes the calculus for checking satisfiability of $(\mathcal{R}, \mathcal{A})$ for $\mathcal{R}$ an admissible set of rules. This algorithm differs from the standard one 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 [3] to ensure termination.

Our derivation rules generate only models of the form 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 (10) 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.

Definition 5. ***Generalized Individuals.*** *Let* T *and* $\mathit{\Gamma}$ *be two disjoint countably infinite sets of* tree *and* graph symbols*. A* generalized individual *is a finite string* $\alpha_0.\alpha_1. \ldots .\alpha_n$ *such that* $\alpha_0 \in N_I$, $\alpha_i \in \mathrm{T} \cup \mathit{\Gamma}$ *for* $1 \leq i \leq n$, *and* $\alpha_{i-1} \in \mathit{\Gamma}$ *implies* $\alpha_i \notin \mathit{\Gamma}$. *If* $\alpha_n \in N_I$, *the individual is* named*; if* $\alpha_n \in \mathrm{T}$, *the individual is a* tree individual*; and if* $\alpha_n \in \mathit{\Gamma}$, *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 all concepts in it are possibly negated atomic concepts, it is nonempty, and it 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 $\mathcal{L}_{\mathcal{A}}(s) = \{C \mid C(s) \in \mathcal{A}$ and C is blocking-relevant $\}$ and $\mathcal{L}_{\mathcal{A}}(s,t) = \{R \mid R(s,t) \in \mathcal{A}\}$. 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; and s is* blocked *iff it is 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 $\mathsf{merge}_{\mathcal{A}}(s \rightarrow t)$ 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 *iff $\bot \in \mathcal{A}$; 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.*

5 From DL Axioms to Graphs

Validating our approach is currently difficult due to the lack of test data. In order to obtain some test data and to facilitate the adoption of our approach, we

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 $\mathcal{A}$.	

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 (1)–(3). 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. 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.

First, 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.

Second, 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 (1)–(3), 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."

Our algorithm is parameterized with a DL TBox $\mathcal{T}_1$, a set of relevant concepts N_{C_g}, and 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 $\lambda\langle i\rangle = \{A\}$, after which 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, α is translated into a graph-regular rule and added 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 ignores α.

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; we refer the interested reader to the long version of this paper [6] for a more detailed discussion. The transformation tool can be downloaded from HermiT's Web site.

We applied our transformation algorithm 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. 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.

6 Evaluation and Discussion

We have implemented our reasoning calculus in the HermiT reasoner. To evaluate the approach, we have classified the original ontologies using HermiT, transformed them using the algorithm from Section 5 into graph-extended KBs, and classified the resulting KBs using our reasoning algorithm 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. These results show that, even with a prototypical implementation, we can process complex ontologies, which we take as indication that our approach is practically feasible.

From a representation point of view, the transformed ontologies are more constrained than the original ones, so we expect to obtain new entailments. In GALEN, we discovered a concept that is satisfiable in the original ontology, but is unsatisfiable in the transformed ontology, which revealed a modeling error; see [6] for more details. 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 $Heart \sqsubseteq Organ$, 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.

7 Conclusion

We have extended DLs with description graphs and rules, which can be used to describe structured objects. Unlike most existing combinations of DLs and rules in which rules can be used only for query answering [5, 8, 9, 1, 7], our rules also fully participate in TBox 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 two 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. Second, to allow for a wider users' community, we would need to extend ontology editors such as Protégé with description graphs.

References

1. 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.
2. I. Horrocks and P. F. Patel-Schneider. A Proposal for an OWL Rules Language. In *Proc. WWW 2004*, pages 723–731, 2004.
3. I. Horrocks, U. Sattler, and S. Tobies. Practical Reasoning for Very Expressive Description Logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.
4. O. Kutz, I. Horrocks, and U. Sattler. The Even More Irresistible SROIQ. In *Proc. KR 2006*, pages 68–78, 2006.
5. A. Y. Levy and M.-C. Rousset. Combining Horn Rules and Description Logics in CARIN. *Artificial Intelligence*, 104(1–2):165–209, 1998.
6. B. Motik, B. Cuenca Grau, and U. Sattler. Structured Objects in OWL: Representation and Reasoning. In *Proc. WWW 2008*, Beijing, China, April 21-25 2008. ACM Press. To appear.
7. B. Motik and R. Rosati. A Faithful Integration of Description Logics with Logic Programming. In *Proc. IJCAI 2007*, pages 477–482, 2007.
8. B. Motik, U. Sattler, and R. Studer. Query Answering for OWL-DL with Rules. *Journal of Web Semantics*, 3(1):41–60, 2005.
9. R. Rosati. $\mathcal{DL}+log$: A Tight Integration of Description Logics and Disjunctive Datalog. In *Proc. KR 2006*, pages 68–78, 2006.
10. 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.