Representing Ontologies Using Description Logics, Description Graphs, and Rules

Boris Motik [a,*], Bernardo Cuenca Grau [a], Ian Horrocks [a], Ulrike Sattler [b]

[a] *University of Oxford, UK*

[b] *University of Manchester, UK*

Abstract

Description logics (DLs) are a family of state-of-the-art knowledge representation languages, and their expressive power has been carefully crafted to provide useful knowledge modeling primitives while allowing for practically effective decision procedures for the basic reasoning problems. Recent experience with DLs, however, has shown that their expressivity is often insufficient to accurately describe *structured objects*—objects whose parts are interconnected in arbitrary, rather than tree-like ways. DL knowledge bases describing structured objects are therefore usually underconstrained, which precludes the entailment of certain consequences and causes performance problems during reasoning.

To address this problem, we propose an extension of DL languages with *description graphs*—a knowledge modeling construct that can accurately describe objects with parts connected in arbitrary ways. Furthermore, to enable modeling the conditional aspects of structured objects, we also extend DLs with rules. We present an in-depth study of the computational properties of such a formalism. In particular, we first identify the sources of undecidability of the general, unrestricted formalism. Based on that analysis, we then investigate several restrictions of the general formalism that make reasoning decidable. We present practical evidence that such a logic can be used to model nontrivial structured objects. Finally, we present a practical decision procedure for our formalism, as well as tight complexity bounds.

Key words: knowledge representation, description logics, structured objects, ontologies

⋆ This is an extended version of two papers published at WWW 2008 [29] and KR 2008 [28], respectively.

* Corresponding author.

Email addresses: boris.motik@comlab.ox.ac.uk (Boris Motik), berg@comlab.ox.ac.uk (Bernardo Cuenca Grau), ian.horrocks@comlab.ox.ac.uk (Ian Horrocks), sattler@cs.man.ac.uk (Ulrike Sattler).

1 Introduction

The Web Ontology Language (OWL) is a well-known language for ontology modeling in the Semantic Web [35]. The World Wide Web Consortium (W3C) is currently working on a revision of OWL—called OWL 2 [10]—whose main goal is to address some of the limitations of OWL. The formal underpinnings of OWL and OWL 2 are provided by description logics (DLs)[3]–knowledge representation formalisms with well-understood formal properties.

DLs are often used to describe *structured objects*—objects whose parts are interconnected in complex ways. Such objects abound in molecular biology and the clinical sciences, and clinical ontologies such as GALEN, the Foundational Model of Anatomy (FMA), and the National Cancer Institute (NCI) Thesaurus describe numerous structured objects. For example, FMA models the human hand as consisting of the fingers, the palm, various bones, blood vessels, and so on, all of which are highly interconnected.

Modeling structured objects poses numerous problems to DLs and the OWL family of languages. The design of DLs has been driven by the desire to provide practically useful knowledge modeling primitives while ensuring decidability of the core reasoning problems. To achieve the latter goal, the modeling constructs available in DLs are usually carefully crafted so that the resulting language exhibits a variant of the *tree-model property* [40]: each satisfiable DL ontology always has at least one model whose elements are connected in a tree-like manner. This property can be used to derive a decision procedure; however, it also prevents one from accurately describing (usually non-tree-like) structured objects since, whenever a model exists, at least one model does not reflect the intended structure. This technical problem has severe consequences in practice [29]. In search of the "correct" way of describing structured objects, modelers often create overly complex descriptions; however, since the required expressive power is actually missing, such descriptions do not entail the consequences that would follow if the descriptions accurately captured the intended structure. We discuss the expressivity limitations of DLs in more detail in Section 3 and present a practically-motivated example.

In order to address this lack of expressivity, in this paper we extend DLs with *description graphs*, which can be understood as schema-level descriptions of structured objects. To allow for the representation of conditional statements about structured objects, we also extend DLs with first-order rules [20]. In this way, we obtain a powerful and versatile knowledge representation formalism. It allows us, for example, to describe the structure of the hand using description graphs, statements such as "if a bone in the hand is fractured, then the hand is fractured as well" using rules, and nonstructural aspects of the domain such as "a medical doctor is a person with an MD degree" using DLs.

Unsurprisingly, this formalism is undecidable in its unrestricted form. It is widely recognized that reasoning algorithms are more likely to be effective in practice if the underlying logics are decidable. Therefore, we discuss the main causes of undecidability and investigate restrictions under which the formalism becomes decidable.

We have observed that structured objects can often be described by a possibly large, yet bounded number of parts. For example, a human body consists of organs all of which can be decomposed into smaller parts; however, further decomposition will eventually lead to parts that one does not want or know how to describe any further. In this vein, FMA describes the skeleton of the hand, but it does not describe the internal structure of the distal phalanges of the fingers. The number of parts needed to describe the hand is therefore determined by the granularity of the hierarchical decomposition of the hand. This decomposition naturally defines an acyclic hierarchy of description graphs. For example, the fingers can be described by description graphs that are subordinate to that of the hand; however, the description graph for the hand is not naturally subordinate to the description graphs for the fingers. We use this observation to define a particular *acyclicity* restriction on description graphs. Acyclicity bounds the number of parts that one needs to reason with, which, provided that there are no DL axioms, can be used to obtain a decision procedure for the basic reasoning problems.

If description graphs are used in combination with DL axioms, the acyclicity condition alone does not ensure decidability due to possible interactions between DL axioms, graphs, and rules [26]. To ensure decidability, we limit this interaction by imposing an additional *role separation* condition. In particular, we separate the roles (i.e., the binary predicates) that can be used in DL axioms from the roles that can be used in rules; furthermore, depending on the expressivity of the DL being used, we may additionally require DL axioms not to refer to the roles used in the description graphs.

We present a hypertableau-based [32] reasoning algorithm that decides the satisfiability problem in the decidable cases, and that acts as a semi-decision procedure for some undecidable ones. Furthermore, we present tight complexity bounds for the decidable variants of our formalism and identify the main sources of complexity. We have implemented the reasoning algorithm in the HermiT [1] reasoner [31], and our initial experiments have shown the algorithm to be amenable to practice.

Evaluation of our approach is currently difficult due to the lack of test data. We have therefore devised an algorithm that extracts description graphs from existing 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 thus derived ontologies agree with their intuition. Our transformation can thus be used as a starting point for a more comprehensive remodeling of ontologies using description graphs. Our experiments already allowed us to discover a modeling error in GALEN, which we take as indication of the practical usefulness of our formalism. 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 DLs.

We believe that description graphs can be used for modeling structured objects in a number of domains, of which we list a few next.

[1] `http://www.hermit-reasoner.com/`

- **Anatomy.** In Sections 3 and 4 we present a comprehensive example of how description graphs can be applied to model human anatomy.
- **Chemistry.** The precise description of molecules is an important problem in bioinformatics [23]. A formal representation of molecules and chemical compounds is often used to integrate information from different chemical databases [23]. The structure of molecules is often not tree-like. For example, hydrocarbons are chemical compounds containing carbon–hydrogen chains, and benzene is a hydrocarbon whose molecules contain exactly one benzene ring. The structure of benzene can be described using our formalism: description graphs can be used to represent the benzene ring (which is bounded in size), while standard OWL axioms can be used to represent tree-like carbon–hydrogen chains.
- **Scientific Workflows.** Scientific workflows are descriptions of the steps of scientific experiments, and they are often represented as directed graphs in which each node depicts a single experiment step and each edge represents information flow between two steps. The precise description of workflows is increasingly important, for example, in bioinformatics. Attempts were made to provide semantics to workflows using OWL [16], but the success has been rather limited so far due to their non-tree-like structure. Since workflows are typically bounded, however, they can naturally be represented using description graphs.
- **Engineering.** OWL has recently been used in engineering domains, such as the aerospace industry, which involve the representation of very complex structured objects such as aeroplanes [19]. The number of parts needed to describe an aircraft is naturally bounded (in the same way as it is in the case of the human body), so such domains can easily be represented using description graphs [17].

2 Preliminaries

The formal underpinnings of OWL 2 are provided by the DL $\mathcal{SROIQ}$ [25]. To make our results easier to follow, however, in this paper we consider the DL $\mathcal{SHOIQ}^+$, which covers all of $\mathcal{SROIQ}$ except for the so-called complex role inclusions. Using a well-known encoding [21,13], complex role inclusions can be encoded using $\mathcal{SHOIQ}^+$ axioms, so the decidability results and reasoning algorithms from this paper can be easily extended to $\mathcal{SROIQ}$ and OWL 2. We present the definition of $\mathcal{SHOIQ}^+$ in Section 2.1. In Section 2.2, we recapitulate the well-known principles for extending DLs with first-order rules. Finally, in Section 2.3 we present an overview of the hypertableau algorithm for $\mathcal{SHOIQ}^+$ [32].

2.1 The Description Logic $\mathcal{SHOIQ}^+$

A $\mathcal{SHOIQ}^+$ *signature* is a triple (N_C, N_R, N_I) consisting of mutually disjoint sets of *atomic concepts* N_C, *atomic roles* N_R, and *named individuals* N_I. In the rest of this paper, we assume that the signature is implicit in all relevant definitions.

A *role* is either R or R^- (*inverse role*), for $R \in N_R$. The function $\mathsf{inv}(\cdot)$ is defined on the set of roles as $\mathsf{inv}(R) = R^-$ and $\mathsf{inv}(R^-) = R$. An *RBox axiom* is an expres-

sion of the form $R_1 \sqsubseteq R_2$ (*role inclusion*), $\mathsf{Dis}(S_1, S_2)$ (*role disjointness*), $\mathsf{Ref}(R)$ (*reflexivity*), $\mathsf{Irr}(S)$ (*irreflexivity*), $\mathsf{Sym}(R)$ (*symmetry*), $\mathsf{Asy}(S)$ (*asymmetry*), and $\mathsf{Tra}(R)$ (*transitivity*), where R, R_1, and R_2 are roles, and S, S_1, and S_2 are *simple* roles, as defined next. For X a set of RBox axioms, let $\sqsubseteq^*_X$ be the reflexive-transitive closure of the relation $\{R_1 \sqsubseteq R_2, \mathsf{inv}(R_1) \sqsubseteq \mathsf{inv}(R_2) \mid R_1 \sqsubseteq R_2 \in X\}$. A role R is *transitive* in X if a role R' exists such that $R' \sqsubseteq^*_X R$, $R \sqsubseteq^*_X R'$, and either $\mathsf{Tra}(R') \in X$ or $\mathsf{Tra}(\mathsf{inv}(R')) \in X$. A role S is *simple* in X if no transitive role R exists such that $R \sqsubseteq^*_X S$.

Given a set of RBox axioms X, the set of *concepts* w.r.t. X is the smallest set containing $\top$ (the *top concept*), $\bot$ (the *bottom concept*), A (*atomic concept*), $\{a\}$ (*nominal*), $\neg C$ (*negation*), $C \sqcap D$ (*conjunction*), $C \sqcup D$ (*disjunction*), $\exists R.C$ (*existential restriction*), $\forall R.C$ (*universal restriction*), $\exists S.\mathsf{Self}$ (*local reflexivity*), $\geq n\, S.C$ (*at-least restriction*), and $\leq n\, S.C$ (*at-most restriction*), where A is an atomic concept, a is an individual, C and D are concepts, R is a role, S is a simple role w.r.t. X, and n is a nonnegative integer. The set of *literal concepts* is defined as $N_L = N_C \cup \{\neg A \mid A \in N_C\}$. A *TBox* $\mathcal{T}$ is a finite set of RBox axioms and *general concept inclusion* (GCI) axioms $C \sqsubseteq D$, where C and D are concepts w.r.t. the subset of the RBox axioms of $\mathcal{T}$.[2]

An *assertion* is an expression of the form $C(a)$ (*concept assertion*), $R(a, b)$ (*role assertion*), $a \approx b$ (*equality assertion*), and $a \not\approx b$ (*inequality assertion*), where C is a concept, R is a role, and a and b are named individuals. An *ABox* $\mathcal{A}$ is a finite set of assertions. Finally, a $\mathcal{SHOIQ}^+$ *knowledge base* is a pair $(\mathcal{T}, \mathcal{A})$ where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox.

An *interpretation* for a signature (N_C, N_R, N_I) is a tuple $I = (\triangle^I, \cdot^I)$, where $\triangle^I$ is a nonempty set called the *interpretation domain* and $\cdot^I$ is a function assigning an element $a^I \in \triangle^I$ to each named individual $a \in N_I$, a set $A^I \subseteq \triangle^I$ to each atomic concept $A \in N_C$, and a relation $R^I \subseteq \triangle^I \times \triangle^I$ to each atomic role $R \in N_R$. The extension of $\cdot^I$ to concepts and roles, and satisfaction of axioms and assertions in I is defined as shown in Table 1. An interpretation I is a *model* of $(\mathcal{T}, \mathcal{A})$, written $I \models (\mathcal{T}, \mathcal{A})$, if and only if all axioms of $\mathcal{T}$ and all assertions of $\mathcal{A}$ are satisfied in I.

The basic inference problem for $\mathcal{SHOIQ}^+$ is checking whether $(\mathcal{T}, \mathcal{A})$ is *satisfiable*—that is, whether a model of $(\mathcal{T}, \mathcal{A})$ exists. A concept C *subsumes* a concept D w.r.t. $(\mathcal{T}, \mathcal{A})$, written $(\mathcal{T}, \mathcal{A}) \models C \sqsubseteq D$, if $C^I \subseteq D^I$ for each model I of $(\mathcal{T}, \mathcal{A})$. It is well known that $(\mathcal{T}, \mathcal{A}) \models C \sqsubseteq D$ if and only if $(\mathcal{T}, \mathcal{A} \cup \{C(a), \neg D(a)\})$ is unsatisfiable, where a is an individual occurring in neither $\mathcal{T}$ nor $\mathcal{A}$ [3].

The *negation-normal form* $\mathsf{nnf}(C)$ of a concept C is the concept equivalent to C in which negations occur only in front of atomic concepts and concepts of the form $\{a\}$ and $\exists S.\mathsf{Self}$. The concept $\mathsf{nnf}(C)$ can be computed in time linear in the size of C [3]. With $|\mathcal{K}|$ we denote the size of a knowledge base $\mathcal{K}$—that is, the number of symbols required to encode $\mathcal{K}$ on the input tape of a Turing machine (numbers can be coded in binary).

[2] The TBox $\mathcal{T}$ is sometimes assumed to contain only GCIs, and all RBox axioms are represented as a separate set $\mathcal{R}$; however, to simplify the presentation in the following sections, in this paper we assume that $\mathcal{T}$ contains GCIs as well as RBox axioms.

Table 1
Model-Theoretic Semantics of $\mathcal{SHOIQ}^+$

Interpretation of Roles and Concepts		
$(R^-)^I$	$=$	$\{\langle y, x\rangle \mid \langle x, y\rangle \in R^I\}$
$\top^I$	$=$	$\triangle^I$
$\bot^I$	$=$	$\emptyset$
$\{s\}^I$	$=$	$\{s^I\}$
$(\neg C)^I$	$=$	$\triangle^I \setminus C^I$
$(C \sqcap D)^I$	$=$	$C^I \cap D^I$
$(C \sqcup D)^I$	$=$	$C^I \cup D^I$
$(\exists R.C)^I$	$=$	$\{x \mid \exists y : \langle x, y\rangle \in R^I \wedge y \in C^I\}$
$(\forall R.C)^I$	$=$	$\{x \mid \forall y : \langle x, y\rangle \in R^I \rightarrow y \in C^I\}$
$(\exists S.\mathsf{Self})^I$	$=$	$\{x \mid \langle x, x\rangle \in S^I\}$
$(\geq n\, S.C)^I$	$=$	$\{x \mid \sharp\{y \mid \langle x, y\rangle \in S^I \wedge y \in C^I\} \geq n\}$
$(\leq n\, S.C)^I$	$=$	$\{x \mid \sharp\{y \mid \langle x, y\rangle \in S^I \wedge y \in C^I\} \leq n\}$
Interpretation of Axioms and Assertions		
$I \models C \sqsubseteq D$	iff	$C^I \subseteq D^I$
$I \models R_1 \sqsubseteq R_2$	iff	$R_1^I \subseteq R_2^I$
$I \models \mathsf{Dis}(S_1, S_2)$	iff	$S_1^I \cap S_2^I = \emptyset$
$I \models \mathsf{Ref}(R)$	iff	$\forall x \in \triangle^I : \langle x, x\rangle \in R^I$
$I \models \mathsf{Irr}(S)$	iff	$\forall x \in \triangle^I : \langle x, x\rangle \notin S^I$
$I \models \mathsf{Sym}(R)$	iff	$R^I \subseteq (\mathsf{inv}(R))^I$
$I \models \mathsf{Asy}(S)$	iff	$S^I \cap (\mathsf{inv}(S))^I = \emptyset$
$I \models \mathsf{Tra}(R)$	iff	$\forall x, y, z \in \triangle^I : \langle x, y\rangle \in R^I \wedge \langle y, z\rangle \in R^I \rightarrow \langle x, z\rangle \in R^I$
$I \models C(a)$	iff	$a^I \in C^I$
$I \models R(a, b)$	iff	$\langle a^I, b^I\rangle \in R^I$
$I \models a \approx b$	iff	$a^I = b^I$
$I \models a \not\approx b$	iff	$a^I \neq b^I$

Note: $\sharp N$ is the number of elements in N.

$\mathcal{SHIQ}^+$ and $\mathcal{SHOQ}^+$ are obtained from $\mathcal{SHOIQ}^+$ by disallowing nominals and inverse roles, respectively. $\mathcal{ALCHOIQ}^+$ is obtained from $\mathcal{SHOIQ}^+$ by disallowing transitivity axioms. $\mathcal{ALCIF}$ allows for negation, conjunction, disjunction, existential and universal restrictions, inverse roles, and axioms of the form $\top \sqsubseteq \leq 1\, R.\top$. Finally, $\mathcal{ALCF}$ is obtained from $\mathcal{ALCIF}$ by disallowing inverse roles.

2.2 Extending DLs with Rules

Description logics can be extended with *rules*—clauses interpreted under standard first-order semantics—in a straightforward way [26,20,14]. Let N_V be a set of *variables* disjoint with the set of individuals N_I. An *atom* is an expression of the form $C(s)$, $R(s, t)$, or $s \approx t$, for s and t individuals or variables, C a concept, and R a role. A *rule* is an expression of the form

$$U_1 \wedge \ldots \wedge U_m \rightarrow V_1 \vee \ldots \vee V_n \tag{1}$$

Table 2
Satisfaction of Rules in an Interpretation

$I, \mu \models C(s)$	iff	$s^{I,\mu} \in C^I$
$I, \mu \models R(s,t)$	iff	$\langle s^{I,\mu}, t^{I,\mu} \rangle \in R^I$
$I, \mu \models s \approx t$	iff	$s^{I,\mu} = t^{I,\mu}$
$I, \mu \models \bigwedge_{i=1}^m U_i \rightarrow \bigvee_{j=1}^n V_j$	iff	$I, \mu \models U_i$ for each $1 \leq i \leq m$ implies $I, \mu \models V_j$ for some $1 \leq j \leq n$
$I \models \bigwedge_{i=1}^m U_i \rightarrow \bigvee_{j=1}^n V_j$	iff	$I, \mu \models \bigwedge_{i=1}^m U_i \rightarrow \bigvee_{j=1}^n V_j$ for all mappings μ
$I \models \mathcal{R}$	iff	$I \models r$ for each rule $r \in \mathcal{R}$

where U_i and V_j are atoms, $m \geq 0$, and $n \geq 0$. The conjunction $U_1 \wedge \ldots \wedge U_m$ is called the *body*, and the disjunction $V_1 \vee \ldots \vee V_n$ is called the *head*. Without loss of generality, we assume that no rule r contains $\approx$ in the body. The empty body and the empty head of a rule are written as $\top$ and $\bot$, respectively. A rule is *Horn* if the head of the rule contains at most one atom. Variables x and y are *directly connected* in a rule r if they occur together in some body atom of r; furthermore, *connected* is the transitive closure of directly connected. A rule r is *connected* if each pair of variables x and y occurring in r is connected in r.

Let $I = (\triangle^I, \cdot^I)$ be an interpretation and $\mu : N_V \rightarrow \triangle^I$ a mapping of variables to elements of the interpretation domain. Let $a^{I,\mu} = a^I$ for an individual a and $x^{I,\mu} = \mu(x)$ for a variable x. Satisfaction of an atom, rule, and a set of rules $\mathcal{R}$ in I and μ is defined in Table 2.

2.3 Hypertableau Calculus for $\mathcal{SHOIQ}^+$

We now present an overview of the hypertableau calculus [32], which can be used to decide the satisfiability of a $\mathcal{SHOIQ}^+$ knowledge base $(\mathcal{T}, \mathcal{A})$.

The algorithm first preprocesses $(\mathcal{T}, \mathcal{A})$ into a set of rules $\Xi_{\mathcal{T}}(\mathcal{T})$ and an ABox $\mathcal{A} \cup \Xi_{\mathcal{A}}(\mathcal{T})$. This transformation consists of three steps. First, transitivity axioms are eliminated from $\mathcal{T}$ by encoding them using general concept inclusions; similar encodings are well known in the context of various description and modal logics [39,38,27]. Second, axioms are normalized and complex concepts are replaced with atomic ones in a way similar to the structural transformation [36]. Third, the normalized axioms are translated into rules by using the correspondences between description and first-order logic [8]. We omit the details of the preprocessing for the sake of brevity; they can be found in [32, Section 4.1]. All steps are satisfiability preserving; thus, $\Xi_{\mathcal{T}}(\mathcal{T})$ and $\mathcal{A} \cup \Xi_{\mathcal{A}}(\mathcal{T})$ are equisatisfiable with $(\mathcal{T}, \mathcal{A})$.

Preprocessing produces HT-rules—syntactically restricted rules on which the hypertableau calculus is guaranteed to terminate. In the definition of HT-rules and in the rest of this paper, we often use the following function ar. Given a role R and variables or constants s and t, this function returns an atom with an atomic role that is semantically equivalent to $R(s,t)$.

$$\mathsf{ar}(R,s,t) = \begin{cases} R(s,t) \text{ if } R \text{ is an atomic role} \\ S(t,s) \text{ if } R \text{ is an inverse role and } R = S^- \end{cases}$$

Definition 1 (HT-Rule)**.** *We assume that the set of atomic concepts N_C contains a* nominal guard concept *O_a for each individual a, and that these concepts do not occur in any input knowledge base.*

An at-most equality *is an atom of the form $s \approx t\,@^{u}_{\leq n\,S.B}$, where s, t, and u are constants or variables, n is a nonnegative integer, S is a role, and B is a literal concept; the part $@^{u}_{\leq n\,S.B}$ of the atom is called the* annotation*. This atom is semantically equivalent to $s \approx t$.*

An HT-rule *r is a rule of the form* (1) *with $m \geq 0$ and $n \geq 0$, in which it must be possible to separate the variables into a* center variable *x, a set of* branch variables *y_i, and a set of* nominal variables *z_j such that the following properties hold, for A an atomic concept, B a literal but not a nominal guard concept, O_a a nominal guard concept, R an atomic role, and S a role.*

- *Each atom in the body of r is of the form $A(x)$, $R(x,x)$, $R(x,y_i)$, $R(y_i,x)$, $A(y_i)$, or $A(z_j)$.*
- *Each atom in the head of r is of the form $B(x)$, $\geq h\,S.B(x)$, $B(y_i)$, $R(x,x)$, $R(x,y_i)$, $R(y_i,x)$, $R(x,z_j)$, $R(z_j,x)$, $x \approx z_j$, or $y_i \approx y_j\,@^{x}_{\leq h\,S.B}$.*
- *Each y_i occurs in the body of r in an atom of the form $R(x,y_i)$ or $R(y_i,x)$.*
- *Each z_j occurs in the body of r in an atom of the form $O_a(z_j)$.*
- *Each equality $y_i \approx y_j\,@^{x}_{\leq h\,S.A}$ in the head of r occurs in a subclause of r of the form* (2) *and no y_k with $1 \leq k \leq h+1$ occurs elsewhere in r.*

$$\text{(2)} \qquad \dots \bigwedge_{k=1}^{h+1} [\mathsf{ar}(S,x,y_k) \wedge A(y_k)] \dots \rightarrow \dots \bigvee_{1 \leq k < \ell \leq h+1} y_k \approx y_\ell\,@^{x}_{\leq h\,S.A} \dots$$

- *Each equality $y_i \approx y_j\,@^{x}_{\leq n\,S.\neg A}$ in the head of r occurs in a subclause of r of the form* (3) *and no y_k with $1 \leq k \leq h+1$ occurs elsewhere in r.*

$$\text{(3)} \qquad \dots \bigwedge_{k=1}^{h+1} \mathsf{ar}(S,x,y_k) \dots \rightarrow \dots \bigvee_{k=1}^{h+1} A(y_k) \vee \bigvee_{1 \leq k < \ell \leq h+1} y_k \approx y_\ell\,@^{x}_{\leq h\,S.\neg A} \dots$$

Intuitively, the body and the head of HT-rules can be seen as being "star-shaped": the variable x represents the center of the star, and the branch variables y_i can be connected to the center only through role atoms. Such a shape ensures that HT-rules can enforce only tree-like models—a property that can be used to explain the good computational properties of many DLs [40].

Atoms of the form $x \approx z_j$ in HT-rules stem from nominals. For example, axiom $C \sqsubseteq \{a\}$ naturally corresponds to the rule $C(x) \rightarrow x \approx a$. To avoid making the calculus unnecessarily complex, however, the rules should not contain constants [32]. Therefore, nominal guard concepts are used to "push" all constants from the rules into the ABox. For example, the mentioned rule is transformed into an HT-rule $C(x) \wedge O_a(z) \rightarrow x \approx z$ and an assertion $O_a(a)$. Nominal guard concepts are unique for the nominal and they are used only internally by the algorithm—that is, they are not allowed to occur in the input knowledge base.

At-most equalities $y_i \approx y_j \, @^x_{\leq n\, R.C}$ stem from the translation of at-most concepts; for example, $\top \sqsubseteq \leq 1\, R.\top$ is translated into $R(x, y_1) \wedge R(x, y_2) \rightarrow y_1 \approx y_2 \, @^x_{\leq 1\, R.\top}$. The *annotation* $@^x_{\leq 1\, R.\top}$ does not affect the meaning of the equality; it merely records its provenance, and we shall discuss its usage shortly. The concept $\exists R.C$ is used in the rest of this paper as an abbreviation for $\geq 1\, R.C$.

The hypertableau calculus takes a set of HT-rules $\mathcal{R}$ and an input ABox $\mathcal{A}$, and it decides the satisfiability of $(\mathcal{R}, \mathcal{A})$.

Definition 2 (Hypertableau Algorithm)**.**

Individuals. *Given a set of* named *individuals* N_I*, the set of* root individuals N_O *is the smallest set such that* $N_I \subseteq N_O$ *and, if* $x \in N_O$*, then* $x.\langle R, B, i\rangle \in N_O$ *for each role* R*, literal concept* B*, and integer* i*. The set of* generalized individuals N_A *is the smallest set such that* $N_O \subseteq N_A$ *and, if* $x \in N_A$*, then* $x.i \in N_A$ *for each integer* i*. The individuals in* $N_A \setminus N_O$ *are* tree individuals*.*

A tree individual $x.i$ *is a* successor *of* x*, and* x *is a* predecessor *of* $x.i$*.* Descendant *and* ancestor *are the transitive closures of successor and predecessor, respectively.*

ABoxes. *The hypertableau algorithm operates on ABoxes that are obtained by extending the standard definition as follows.*

- *In addition to standard assertions, an ABox can contain at-most equalities and a special assertion* $\bot$ *that is false in all interpretations. Furthermore, assertions can refer to the individuals from* N_A *and not only from* N_I*.*
- *Each (in)equality* $s \approx t$ *(*$s \not\approx t$*) also stands for the symmetric (in)equality* $t \approx s$ *(*$t \not\approx s$*). The same is true for annotated at-most equalities.*
- *An ABox* $\mathcal{A}$ *can contain* renamings *of the form* $a \mapsto b$ *where* a *and* b *are root individuals. The relation* $\mapsto$ *in* $\mathcal{A}$ *must be acyclic,* $\mathcal{A}$ *can contain at most one renaming* $a \mapsto b$ *for an individual* a*, and, if* $\mathcal{A}$ *contains* $a \mapsto b$*, then* a *should not occur in any assertion or (in)equality in* $\mathcal{A}$*. An individual* b *is the* canonical name *of a root individual* a *in* $\mathcal{A}$*, written* $b = \|a\|_{\mathcal{A}}$*, iff* $a \mapsto^* b$ *and there exists no individual* $c \neq b$ *such that* $b \mapsto^* c$*, where* $\mapsto^*$ *is the reflexive-transitive closure of* $\mapsto$ *in* $\mathcal{A}$*.*

An input ABox *contains only named individuals, no at-most equalities, no renamings, and in which all concepts are literal and all roles are atomic.*

Satisfaction of such ABoxes in an interpretation is obtained by a straightforward generalization of the definitions in Section 2.1: all individuals are interpreted as elements of the interpretation domain $\triangle^I$*, and* $I \models a \mapsto b$ *iff* $a^I = b^I$*.*

Merge Target. *An individual* t *is a* merge target *for an individual* s *if* t *is a named individual, or* t *is a root individual and* s *is not a named individual, or* s *is a descendant of* t*.*

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

Merging. *The ABox* $\mathsf{merge}_{\mathcal{A}}(s \rightarrow t)$ *is obtained from* $\mathsf{prune}_{\mathcal{A}}(s)$ *by replacing the individual* s *with the individual* t *in all assertions and their annotations (but not in renamings) and, if both* s *and* t *are root individuals, adding the renaming* $s \mapsto t$*.*

Pairwise Anywhere Blocking. *The* labels of an individual s *and* of an individual pair $\langle s, t\rangle$ *in an ABox* $\mathcal{A}$ *are defined as follows:*

$$\mathcal{L}_{\mathcal{A}}(s) = \{\ A \mid A(s) \in \mathcal{A} \text{ and } A \text{ is an atomic concept}\ \}$$
$$\mathcal{L}_{\mathcal{A}}(s,t) = \{\ R \mid R(s,t) \in \mathcal{A}\ \}$$

Let $<$ *be a strict ordering (i.e., a transitive and irreflexive relation) on* N_A *containing the ancestor relation—that is, if* s' *is an ancestor of* s*, then* $s' < s$*. By induction on* $<$*, we assign to each individual* s *in* $\mathcal{A}$ *a status as follows:*

- *a tree individual* s *is* directly blocked by *a tree individual* t *iff, for* s' *and* t' *the predecessors of* s *and* t*, respectively,*
  - t *is not blocked,*
  - $t < s$*,*
  - $\mathcal{L}_{\mathcal{A}}(s) = \mathcal{L}_{\mathcal{A}}(t)$ *and* $\mathcal{L}_{\mathcal{A}}(s') = \mathcal{L}_{\mathcal{A}}(t')$*, and*
  - $\mathcal{L}_{\mathcal{A}}(s,s') = \mathcal{L}_{\mathcal{A}}(t,t')$ *and* $\mathcal{L}_{\mathcal{A}}(s',s) = \mathcal{L}_{\mathcal{A}}(t',t)$*;*
- s *is* indirectly blocked *iff it has a predecessor that is blocked; and*
- s *is* blocked *iff it is either directly or indirectly blocked.*

Derivation Rules. *Table 3 specifies* derivation rules *that, given an ABox* $\mathcal{A}$ *and a set of HT-rules* $\mathcal{R}$*, derive the ABoxes* $\mathcal{A}_1, \ldots, \mathcal{A}_n$*. In the Hyp-rule,* σ *is a mapping from the set of variables in the HT-rule to the individuals occurring in the assertions of* $\mathcal{A}$*, and* $\sigma(U)$ *is the result of replacing each variable* x *in the atom* U *with* $\sigma(x)$*. In the NI-rule, for* u *a root individual,* R *a role,* B *a literal concept, and* i *an integer, we define* $\mathsf{rootfor}(u, R, B, i) = u.\langle R, B, i\rangle$*.* [3]

Rule Precedence. *The* $\approx$*-rule can be applied to a (possibly annotated) equality* $s \approx t$ *in an ABox* $\mathcal{A}$ *only if* $\mathcal{A}$ *does not contain an equality* $s \approx t\,@^{u}_{\leq n\,R.B}$ *to which the NI-rule is applicable.*

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

Derivation. *For a set of HT-rules* $\mathcal{R}$ *and an ABox* $\mathcal{A}$*, a* derivation *is a pair* (T, ρ) *where* T *is a finitely branching tree and* ρ *is a function that labels the nodes of* T *with ABoxes such that the following properties hold for each node* $t \in T$*:*

- $\rho(t) = \mathcal{A}$ *if* t *the root of* T*;*
- t *is a leaf of* T *if* $\bot \in \rho(t)$ *or no derivation rule is applicable to* $\rho(t)$ *and* $\mathcal{R}$*;*
- t *has children* $t_1, \ldots, t_n$ *such that* $\rho(t_1), \ldots, \rho(t_n)$ *are exactly the results of applying one (arbitrarily chosen, but respecting the rule precedence) applicable rule to* $\rho(t)$ *and* $\mathcal{R}$ *in all other cases.*

A derivation is successful *if* T *contains a branch* $t_1, t_2, \ldots$ *such that each ABox* $\rho(t_i)$ *is clash-free.*

[3] The function rootfor is not used in the formalization of the algorithm in [32], and it has been introduced here to make the presentation of the algorithm in Section 6 easier.

Table 3
Derivation Rules of the Hypertableau Calculus

Hyp-rule	If 1. $r \in \mathcal{R}$ with $r = U_1 \wedge \ldots \wedge U_m \to V_1 \vee \ldots \vee V_n$, and 2. a mapping σ from variables of r to the individuals of $\mathcal{A}$ exists such that 2.1 $\sigma(x)$ is not indirectly blocked for each variable $x \in N_V$, 2.2 $\sigma(U_i) \in \mathcal{A}$ for each $1 \leq i \leq m$, and 2.3 $\sigma(V_j) \notin \mathcal{A}$ for each $1 \leq j \leq n$, then $\mathcal{A}_1 := \mathcal{A} \cup \{\bot\}$ if $n = 0$; $\mathcal{A}_j := \mathcal{A} \cup \{\sigma(V_j)\}$ for $1 \leq j \leq n$ otherwise.
$\geq$-rule	If 1. $\geq n\, R.B(s) \in \mathcal{A}$, 2. s is not blocked in $\mathcal{A}$, and 3. $\mathcal{A}$ does not contain individuals $u_1, \ldots, u_n$ such that 3.1 $\{\mathsf{ar}(R, s, u_i), B(u_i) \mid 1 \leq i \leq n\} \cup \{u_i \not\approx u_j \mid 1 \leq i < j \leq n\} \subseteq \mathcal{A}$, and 3.2 u_i is not indirectly blocked in $\mathcal{A}$ for each $1 \leq i \leq n$ then $\mathcal{A}_1 := \mathcal{A} \cup \{\mathsf{ar}(R, s, t_i),\ B(t_i) \mid 1 \leq i \leq n\} \cup \{t_i \not\approx t_j \mid 1 \leq i < j \leq n\}$ where $t_1, \ldots, t_n$ are fresh distinct tree successors of s.
$\approx$-rule	If 1. $s \approx t \in \mathcal{A}$ (the equality can possibly be annotated), 2. $s \neq t$, and 3. neither s not t is indirectly blocked then $\mathcal{A}_1 := \mathsf{merge}_{\mathcal{A}}(s \to t)$ if t is merge target for s, and $\mathcal{A}_1 := \mathsf{merge}_{\mathcal{A}}(t \to s)$ otherwise.
$\bot$-rule	If $s \not\approx s \in \mathcal{A}$ or $\{A(s), \neg A(s)\} \subseteq \mathcal{A}$ where s is not indirectly blocked then $\mathcal{A}_1 := \mathcal{A} \cup \{\bot\}$.
NI-rule	If 1. $s \approx t\, @^{u}_{\leq n\, R.B} \in \mathcal{A}$ (the symmetry of $\approx$ applies as usual), 2. u is a root individual, 3. s is neither a root individual nor a tree successor of u, 4. t is not a root individual, and 5. neither s nor t is indirectly blocked then $\mathcal{A}_i := \mathsf{merge}_{\mathcal{A}}(s \to \lVert\mathsf{rootfor}(u, R, B, i)\rVert_{\mathcal{A}})$ for each $1 \leq i \leq n$.

The *Hyp*-rule is similar to the one of the hypertableau calculus for first order logic [6]: given an HT-rule of the form (1) and an ABox $\mathcal{A}$, the *Hyp*-rule tries to unify the atoms $U_1, \ldots, U_m$ with a subset of the assertions in $\mathcal{A}$; if a unifier σ is found, the rule nondeterministically derives $\sigma(V_j)$ for some $1 \leq j \leq n$. For example, given $R(x, y) \to \exists R.C(x) \vee D(y)$ and an assertion $R(a, b)$, the *Hyp*-rule derives either $\exists R.C(a)$ or $D(b)$. The $\geq$-rule deals with existential quantifiers; for example, given $\exists R.C(a)$, the rule introduces a fresh individual t and derives $R(a, t)$ and $C(t)$. The $\approx$-rule deals with equality; for example, given $a \approx b$, the rule replaces the individual a in all assertions with the individual b, and it introduces a *renaming* $a \mapsto b$ in order to keep track of the merge. As discussed in [32], renamings are necessary to ensure soundness of the *NI*-rule. Finally, the $\bot$-rule detects contradictions such as $A(a)$ and $\neg A(a)$, or $a \not\approx a$.

Termination of the calculus is ensured through *blocking*, the correctness of which relies on the notion of *forest-shaped ABoxes*. Such an ABox is shown in Figure 1, where nodes and edges correspond to individuals and role assertions, respectively. *Named* individuals (shown as black nodes) originate from the *input ABox*, and they can be connected in arbitrary ways. *Tree* individuals (shown as white nodes and

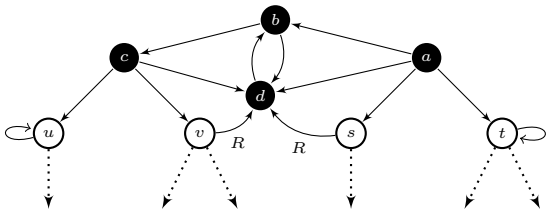


Fig. 1. A Forest-Shaped ABox

called *blockable* individuals in [32]) are introduced by the $\geq$-rule, and they can be connected either to arbitrary named individuals, or to other tree individuals in a *tree-like* way. For convenience, tree individuals are represented in the algorithm as strings; for example, $s = a.1$ denotes that s is the first *successor* of a. Now it is possible to show that the $\geq$-rule can be applied only to nonblocked individuals without jeopardizing completeness. Intuitively, if s is blocked by t in an ABox $\mathcal{A}$ and no derivation rule is applicable to $\mathcal{A}$, then a model of the knowledge base can be constructed by "unraveling" $\mathcal{A}$—that is, by replicating the fragment between s and t infinitely often. Effectively, blocking means that we do not need to introduce tree successors in order to satisfy assertions of the form $\geq n\, R.C(s)$ because we can "reuse" the successors of t.

Applications of most derivation rules preserve the forest shape of an ABox; however, inverse roles, nominals, and number restrictions cause subtle problems. Consider again Figure 1 and assume that d must satisfy an at-most restriction $\leq 1\, R^-.\top$. This implies $v \approx s$, so one individual should be merged into the other; however, this can compromise the tree shape of the ABox. The *NI*-rule deals with this problem by promoting one of v or s into a *root individual*: such individuals can be connected in arbitrary ways even if they do not occur in the input ABox. Thus, an application of the *Hyp*-rule to the ABox in Figure 1 and the HT-rule $R(y_1, x) \wedge R(y_2, x) \rightarrow y_1 \approx y_2\, @^x_{\leq 1\, R^-.\top}$ derives the at-most equality $v \approx s\, @^d_{\leq 1\, R^-.\top}$. By examining the annotation on the equality, the *NI*-rule can detect that the equality stems from an at-most concept, in response to which it turns either v or s into a root individual. It is possible to establish a bound on the number of the introduced root individuals and thus prove termination.

The complexity of the hypertableau calculus is due to a possible interaction between number restrictions, inverse roles, and nominals. If a description logic does not support at least one of these constructors, then the HT-rules in $\Xi_{\mathcal{T}}(\mathcal{T})$ have a simpler form, which prevents the derivation of assertions that satisfy the preconditions of the *NI*-rule. In such cases, each atom of the form $y_i \approx y_j\, @^x_{\leq n\, R.B}$ can be simplified to $y_i \approx y_j$, and the set of root individuals becomes the same as the set of named individuals. Furthermore, Condition 3.2 of the $\geq$-rule is always satisfied, so it need not be explicitly checked.

On $\mathcal{SHOQ}^+$, the HT-rules in $\Xi_{\mathcal{T}}(\mathcal{T})$ have the following simpler form.

Definition 3 (Simple HT-Rule)**.** *A* simple HT-rule *is a rule r of the form* (1) *in which it must be possible to separate the variables into a* center variable *x, a set of*

branch variables y_i, *and a set of* nominal variables z_j *such that the following properties hold, for* A *an atomic concept,* B *a literal but not a nominal guard concept,* O_a *a nominal guard concept, and* R *an atomic role.*

- *Each atom in the body of* r *is of the form* $A(x)$, $R(x, x)$, $R(x, y_i)$, $A(y_i)$, *or* $A(z_j)$.
- *Each atom in the head of* r *is of the form* $B(x)$, $\geq n\, R.B(x)$, $B(y_i)$, $R(x, x)$, $R(x, y_i)$, $R(x, z_j)$, $x \approx z_j$, *or* $y_i \approx y_j$.
- *Each* y_i *occurs in the body of* r *in an atom of the form* $R(x, y_i)$.
- *Each* z_j *occurs in the body of* r *in an atom of the form* $O_a(z_j)$.

Simple HT-rules allow for a simpler version of blocking: instead of pairs of individuals, one needs to consider only single individuals.

Definition 4. ***Single Anywhere Blocking.*** *By induction on* $<$, *each individual* s *in* $\mathcal{A}$ *is assigned a status as follows:*

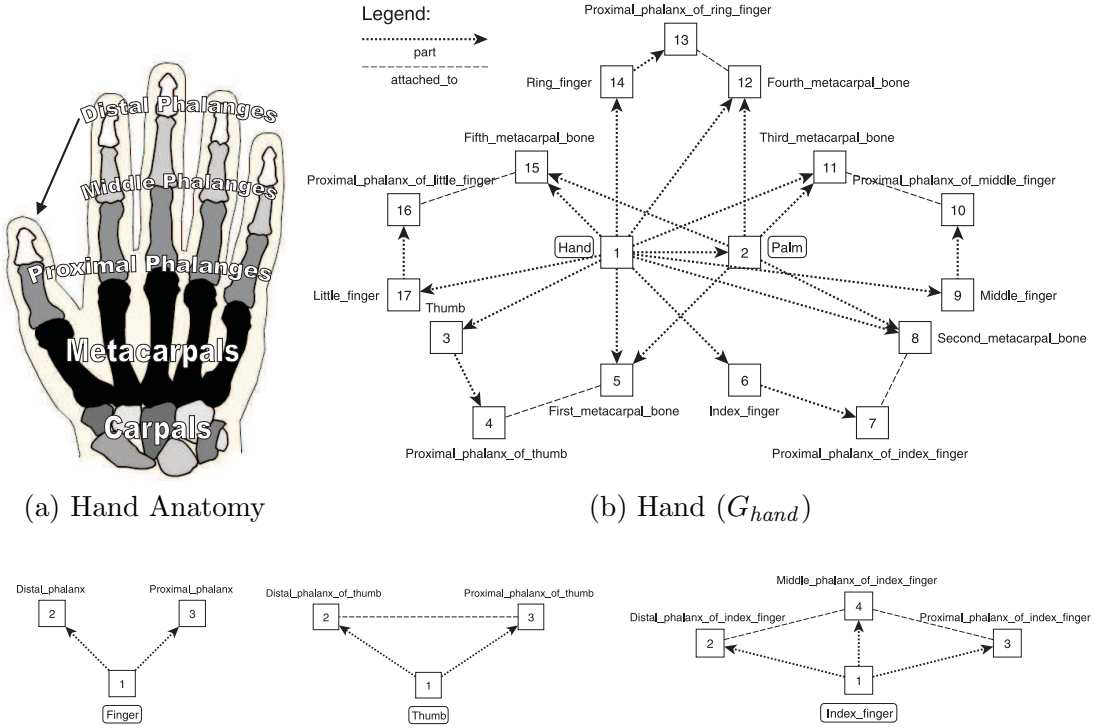
- *a tree individual* s *is* directly blocked by *a tree individual* t *if* t *is not blocked,* $t < s$, *and* $\mathcal{L}_{\mathcal{A}}(s) = \mathcal{L}_{\mathcal{A}}(t)$;
- s *is* indirectly blocked *if it has a predecessor that is blocked; and*
- s *is* blocked *if it is either directly or indirectly blocked.*

3 Problems with Modeling Complex Structures

To understand the limitations of modeling structured objects in DLs (and hence in OWL as well), consider the problem of modeling the skeleton of the human hand, whose structure is shown in Figure 2a. The carpal bones form the base of the hand. The central part of the hand contains the metacarpal bones, one leading to each finger. The fingers consist of phalanges: the proximal phalanges are connected to the metacarpal bones, and all fingers apart from the thumb contain a middle phalanx between the proximal and the distal phalanx. This structure can be intuitively conceptualized as shown in Figures 2b–2e. [4]

This structure can be straightforwardly encoded in a DL ABox $\mathcal{A}$. ABox assertions, however, represent concrete data; thus, $\mathcal{A}$ would represent the structure of *one particular* hand. 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* hands. We should be able to instantiate such a description many times. For example, if we say that each patient has a hand, then for each concrete patient we should instantiate a *different* hand, each with the structure as shown in Figures 2b–2e; depending on the properties of the patient and the axioms in the ontology, each such structure can then exhibit distinct features. This clearly cannot be achieved using ABox assertions.

[4] The relationship *attached_to* is assumed to be bidirectional, so the edges labeled with it are not oriented.



(a) Hand Anatomy (b) Hand (G_{hand})

(c) Finger (G_{finger}) (d) Thumb (G_{thumb}) (e) Index (G_{index_finger})

Fig. 2. The Anatomy of the Hand and its Conceptual Models

We can give a logical, schema-level interpretation to Figures 2b–2e by treating vertices as concepts and arrows as *participation constraints* between the concepts. For example, vertices 1 and 6 would correspond to concepts *Hand* and *Index_finger*, whose instances would be all hands and all index fingers, respectively. Furthermore, the arrow from 1 to 6 would be interpreted as a statement that each hand has an index finger as its part. In DLs, such a participation constraint would commonly be represented by axioms (4)–(5).

$$Hand \sqsubseteq \exists part.Index\_finger \tag{4}$$

$$Hand \sqsubseteq \leq 1\, part.Index\_finger \tag{5}$$

Thus, the knowledge base $\mathcal{K}$ containing axioms (6)–(19) would provide a formalization of the structure shown in Figure 2e. [5]

$$Index\_finger \sqsubseteq \exists part.Distal\_phalanx\_oif \tag{6}$$

$$Index\_finger \sqsubseteq \exists part.Middle\_phalanx\_oif \tag{7}$$

$$Index\_finger \sqsubseteq \exists part.Proximal\_phalanx\_oif \tag{8}$$

$$Distal\_phalanx\_oif \sqsubseteq \exists attached\_to.Middle\_phalanx\_oif \tag{9}$$

$$Middle\_phalanx\_oif \sqsubseteq \exists attached\_to.Distal\_phalanx\_oif \tag{10}$$

[5] The suffix *_of_index_finger* has been abbreviated to *_oif*.

$$\text{(11)} \quad \mathit{Middle\_phalanx\_oif} \sqsubseteq \exists \mathit{attached\_to}.\mathit{Proximal\_phalanx\_oif}$$

$$\text{(12)} \quad \mathit{Proximal\_phalanx\_oif} \sqsubseteq \exists \mathit{attached\_to}.\mathit{Middle\_phalanx\_oif}$$

$$\text{(13)} \quad \mathit{Index\_finger} \sqsubseteq \leq 1\, \mathit{part}.\mathit{Distal\_phalanx\_oif}$$

$$\text{(14)} \quad \mathit{Index\_finger} \sqsubseteq \leq 1\, \mathit{part}.\mathit{Middle\_phalanx\_oif}$$

$$\text{(15)} \quad \mathit{Index\_finger} \sqsubseteq \leq 1\, \mathit{part}.\mathit{Proximal\_phalanx\_oif}$$

$$\text{(16)} \quad \mathit{Distal\_phalanx\_oif} \sqsubseteq \leq 1\, \mathit{attached\_to}.\mathit{Middle\_phalanx\_oif}$$

$$\text{(17)} \quad \mathit{Middle\_phalanx\_oif} \sqsubseteq \leq 1\, \mathit{attached\_to}.\mathit{Distal\_phalanx\_oif}$$

$$\text{(18)} \quad \mathit{Middle\_phalanx\_oif} \sqsubseteq \leq 1\, \mathit{attached\_to}.\mathit{Proximal\_phalanx\_oif}$$

$$\text{(19)} \quad \mathit{Proximal\_phalanx\_oif} \sqsubseteq \leq 1\, \mathit{attached\_to}.\mathit{Middle\_phalanx\_oif}$$

Let I be an interpretation corresponding to Figure 2e in the obvious way. Clearly, I satisfies $\mathcal{K}$, which justifies the formalization of Figure 2e using $\mathcal{K}$.

Let us extend $\mathcal{K}$ with knowledge about bone fractures. For example, let $\mathcal{K}'$ be an extension of $\mathcal{K}$ with axiom (20) stating that, if the middle phalanx of the index finger is broken, then the index finger is broken as well:

$$\text{(20)} \quad \mathit{Index\_finger} \sqcap \exists \mathit{part}.(\mathit{Middle\_phalanx\_oif} \sqcap \mathit{Broken}) \sqsubseteq \mathit{Broken}$$

Given the structure of the index finger shown in Figure 2e, we might expect $\mathcal{K}'$ to imply that if the index finger has a distal phalanx that is attached to a broken middle phalanx, then the index finger is broken as well:

$$\text{(21)} \quad \begin{aligned} \mathit{Index\_finger} \sqcap \exists \mathit{part}.(\mathit{Distal\_phalanx\_oif} \sqcap \\ \exists \mathit{attached\_to}.(\mathit{Middle\_phalanx\_oif} \sqcap \mathit{Broken})) \sqsubseteq \mathit{Broken} \end{aligned}$$

Unfortunately, $\mathcal{K}'$ is underconstrained, and some models of $\mathcal{K}'$ do not correspond to the structure of the index finger shown in Figure 2e. Axioms (7) and (9) both imply the existence of middle phalanges of the index finger, but $\mathcal{K}'$ does not capture the fact that, for any given index finger, these two middle phalanges must be the same object. Thus, the infinite interpretation I' shown in Figure 3 is also a model of $\mathcal{K}'$. In I', even if the middle phalanx of the index finger is broken, the middle phalanges at the second level of the model need not be broken; hence, axiom (20) does not necessarily derive that the index finger is broken and, consequently, axiom (21) is not a consequence of $\mathcal{K}'$.

That $\mathcal{K}'$ is underconstrained can also lead to problems with the performance of reasoning. In order to disprove an entailment, a DL reasoner will try to construct a "canonical" model of $\mathcal{K}'$—that is, a model that contains as little information derivable from $\mathcal{K}'$ as possible. Such models, however, are often highly repetitive and much larger than the intended ones, so constructing them can be costly. The interpretation I' is an example of such a "canonical" model, and it contains an infinite tree of phalanges instead of a finite structure shown in Figure 2e. In our experience, this is the main reason why DL reasoners cannot process complex ontologies such as FMA and certain versions of GALEN.

These problems could be addressed by ensuring that *all* models of $\mathcal{K}'$ resemble as much as possible the intended conceptualization shown in Figures 2b–2e. DL

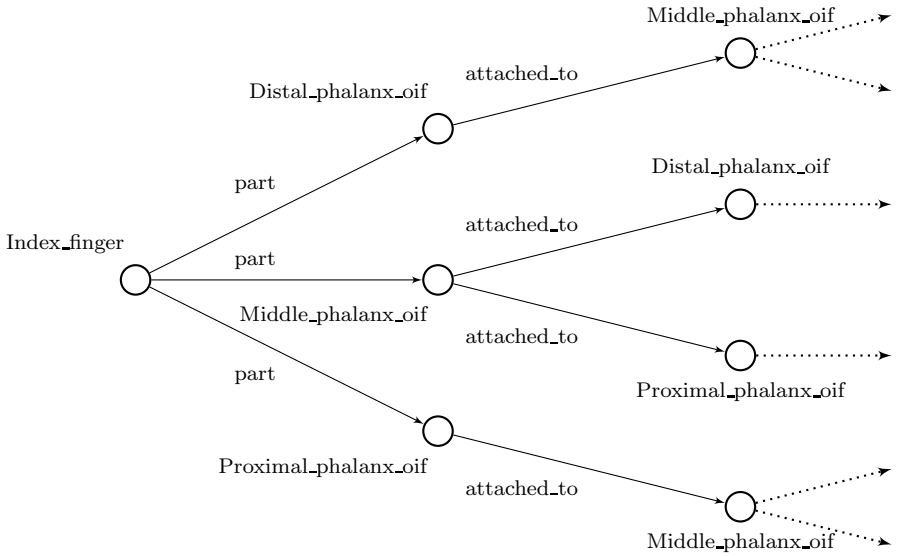


Fig. 3. Unintended Model I'

axioms, however, are usually syntactically restricted such that the resulting logic exhibits (a variant of) the *tree model property* [40]: whenever a DL knowledge base has a model, it has a model of a certain tree shape (such as I'). The tree model property is generally considered desirable because its absence often leads to the undecidability of reasoning. At the same time, however, it also means that we must leave the confines of DLs if we want to faithfully represent structured objects.

Rule formalisms such as datalog [1] can routinely express conditions over nontree structures; however, they typically do not provide for existential quantification. Such rules can thus be applied only to the individuals explicitly mentioned in a knowledge base and cannot express schema constraints such as "each patient has some (unknown) hand." Ontology languages such as OWL-Flight [12], Telos [34], and the logic programming variant of F-Logic [22] are based on datalog and therefore share its restrictions.

Combining rules with description logics overcomes the limitations of datalog and yields a very expressive knowledge representation formalism capable of axiomatizing nontree structures [20,26]. Similarly, the first-order version of F-Logic [22] provides a combination of existential quantification with rules. Such solutions, however, are quite complex and susceptible to modeling errors. Furthermore, the extension of DLs with rules is undecidable even for very simple DLs [26], and the same is the case for the first-order version of F-Logic.

A number of decidable combinations of DLs and rules have been proposed in practice, and decidability is typically achieved by syntactically restricting the applicability of the rules. For example, DL-safe rules [33] are restricted such that they apply only to the explicitly named objects. Role-safe [26] and weakly safe [37] 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 DLs [15,37,30]. Consequently, such extensions are useful mainly for query answering, but not for schema modeling.

To address the problems outlined in this section, $\mathcal{SROIQ}$ [25] provides *complex role inclusions*—axioms of the form $R_1 \circ \ldots \circ R_n \sqsubseteq S$, where $\circ$ stands for role composition. Such axioms are restricted in a way that ensures decidability of the basic reasoning problems. The use of complex role inclusions solves some of the identified problems; however, they still cannot axiomatize arbitrary structures such as the one in Figure 2b.

In the rest of this paper, we propose a formalism for modeling arbitrarily connected structured objects by extending DLs with *description graphs*. For example, Figure 2d can be seen as a description graph showing that each thumb has a proximal and a distal phalanx that are attached to each other. Different structured objects can be represented using separate description graphs, which can be appropriately connected. For example, the hand and the thumb can be represented using two different description graphs, which are connected to each other. Furthermore, structured objects often need to be modeled at different levels of abstraction. For example, we would like to describe the abstract structure common to all fingers as shown in Figure 2c, and then specialize it for, say, the index finger by introducing the middle phalanx as shown in Figure 2e. To this end, our formalism provides for graph specialization statements, which can represent the fact that one structure is more general than another. Finally, in order to represent conditional aspects of the domain, we also allow for arbitrary rules over the description graphs; for example, we can state that, if a bone in the hand is fractured, then the hand is fractured as well. We introduce the formalism in the following section, and show how it can be used to exclude unintended infinite models such as the one in Figure 3.

Our formalism is related to weakly guarded tuple generating dependencies [9] and the guarded fragment of first-order logic [2], which allow for axiomatizing nontree structures of bounded treewidth. Unlike these formalisms, however, graph-extended knowledge bases allow for functional roles and arbitrary rules; furthermore, we present different syntactic restrictions to achieve decidability of reasoning.

4 A Formalism for Modeling Complex Structures

In this section, we present our knowledge representation formalism. We start by defining the notion of a description graph.

Definition 5 (Description Graph)**.** *An ℓ-ary* description graph $G = (V, E, \lambda, M)$ *is a directed labeled graph where*

- $V = \{1, \ldots, \ell\}$ *is a set of* ℓ vertices*,*
- $E \subseteq V \times V$ *is a set of* edges*,*
- λ *is a labeling function that assigns a set of literal concepts* $\lambda\langle i\rangle \subseteq N_L$ *to each vertex* $i \in V$ *and a set of atomic roles* $\lambda\langle i, j\rangle \subseteq N_R$ *to each edge* $\langle i, j\rangle \in E$*, and*
- $M \subseteq N_C$ *is a set of* main concepts *for* G*.*

For A an atomic concept, V_A is the set of vertices that contain A in their label; that is, $V_A = \{k \in V \mid A \in \lambda\langle k\rangle\}$.

We define the vertices of G to be integers in order to be able to use them as indices. The main concepts of G identify the objects whose structure is defined by G. In Figure 2, main concepts are framed with rounded rectangles. Thus, the main concepts for the description graph shown in Figure 2b are *Hand* and *Palm*, meaning that this graph defines the structure of all hands and palms. Intuitively, an instance of a main concept implies the existence of the corresponding *graph instance* of G.

As a notational convenience, we sometimes use $i \xrightarrow{R} j$ to denote that a description graph contains an R-labeled edge from a vertex i to a vertex j.

In order to represent conditions over the structure of a graph, our formalism allows for graph rules. The following definition refines the general notion of a rule introduced in Section 2.2.

Definition 6 (Graph Rule)**.** *A* graph atom *is an atom of the form $G(t_1, \ldots, t_\ell)$, where G is an ℓ-ary description graph and $t_i \in N_I \cup N_V$ for $1 \leq i \leq \ell$. A* graph rule *is a rule in which all concepts and roles in atoms are atomic, and whose head and body can contain graph atoms.*

Next, we introduce graph specializations, which allow us to represent structured objects at different levels of abstraction. For example, we can capture the abstract structure common to all fingers by a description graph G_{finger} shown in Figure 2c, and we can specialize this structure for the thumb by introducing a description graph G_{thumb} shown in Figure 2d. The following graph specialization axiom captures the relationship between these two description graphs:

$$G_{finger} \lhd G_{thumb} \tag{22}$$

Definition 7 (Graph Specialization)**.** *A* graph specialization *is an axiom of the form $G_1 \lhd G_2$, where $G_1 = (V_1, E_1, \lambda_1, M_1)$ and $G_2 = (V_2, E_2, \lambda_2, M_2)$ are description graphs with $V_1 \subseteq V_2$.*

Next, we introduce axioms that allow us to appropriately connect graph instances. For example, the description graph G_{hand} shown in Figure 2b contains the vertices 3 and 4 that represent the thumb and its proximal phalanx, which correspond to the vertices 1 and 3 of the description graph G_{thumb} shown in Figure 2d. We can specify this correspondence using the following *graph alignment*:

$$G_{hand}[3, 4] \leftrightarrow G_{thumb}[1, 3] \tag{23}$$

This axiom ensures that, whenever two instances of G_{hand} and G_{thumb} share the thumb vertex, they share a proximal phalanx vertex as well, and vice versa.

Definition 8 (Graph Alignment)**.** *A* graph alignment *is an axiom of the form $G_1[v_1, \ldots, v_n] \leftrightarrow G_2[w_1, \ldots w_n]$, where G_1 and G_2 are description graphs with sets of vertices V_1 and V_2, respectively, and $v_i \in V_1$ and $w_i \in V_2$ for $1 \leq i \leq n$.*

Finally, we define GBoxes and graph-extended knowledge bases.

Table 4
Satisfaction of GBox Elements in an Interpretation

$I \models G$ for $G = (V, E, \lambda, M)$ iff
Key property:
$\forall x_1, \ldots, x_\ell, y_1, \ldots, y_\ell \in \triangle^I$:
$\langle x_1, \ldots, x_\ell \rangle \in G^I \wedge \langle y_1, \ldots, y_\ell \rangle \in G^I \wedge \bigvee_{1 \leq i \leq \ell} x_i = y_i \rightarrow \bigwedge_{1 \leq j \leq \ell} x_j = y_j$
Disjointness property:
$\forall x_1, \ldots, x_\ell, y_1, \ldots, y_\ell \in \triangle^I : \langle x_1, \ldots, x_\ell \rangle \in G^I \wedge \langle y_1, \ldots, y_\ell \rangle \in G^I \rightarrow \bigwedge_{1 \leq i < j \leq \ell} x_i \neq y_j$
Start property: for each atomic concept $A \in M$,
$\forall x \in \triangle^I : x \in A^I \rightarrow \exists x_1, \ldots, x_\ell \in \triangle^I : \langle x_1, \ldots, x_\ell \rangle \in G^I \wedge \bigvee_{k \in V_A} x = x_k$
Layout property:
$\forall x_1, \ldots, x_\ell \in \triangle^I : \langle x_1, \ldots, x_\ell \rangle \in G^I \rightarrow \bigwedge_{i \in V,\, B \in \lambda\langle i \rangle} x_i \in B^I \wedge \bigwedge_{\langle i,j \rangle \in E,\, R \in \lambda\langle i,j \rangle} \langle x_i, x_j \rangle \in R^I$
$I \models G_1 \lhd G_2$ iff
$\forall x_1, \ldots, x_{\ell_2} \in \triangle^I : \langle x_1, \ldots, x_{\ell_1}, \ldots, x_{\ell_2} \rangle \in G_2^I \rightarrow \langle x_1, \ldots, x_{\ell_1} \rangle \in G_1^I$
$I \models G_1[v_1, \ldots, v_n] \leftrightarrow G_2[w_1, \ldots w_n]$ iff, for each $1 \leq i \leq n$,
$\forall x_1, \ldots, x_{\ell_1}, y_1, \ldots, y_{\ell_2} \in \triangle^I$:
$\langle x_1, \ldots, x_{\ell_1} \rangle \in G_1^I \wedge \langle y_1, \ldots, y_{\ell_2} \rangle \in G_2^I \wedge x_{v_i} = y_{w_i} \rightarrow \bigwedge_{1 \leq j \leq n} x_{v_j} = y_{w_j}$
Note: $\ell_{(i)}$ is the arity of the description graph $G_{(i)}$.

Definition 9 (Formalism)**.** *A* graph box *(GBox) is a tuple* $\mathcal{G} = (\mathcal{G}_G, \mathcal{G}_S, \mathcal{G}_A)$ *where* $\mathcal{G}_G$*,* $\mathcal{G}_S$*, and* $\mathcal{G}_A$ *are finite sets of description graphs, graph specializations over* $\mathcal{G}_G$*, and graph alignments over* $\mathcal{G}_G$*, respectively. The definition of ABoxes from Section 2.1 is extended to allow for* graph assertions *of the form* $G(a_1, \ldots, a_\ell)$ *where* G *is an* ℓ*-ary graph and each* a_i*,* $1 \leq i \leq \ell$*, is an individual. A* graph-extended knowledge base *is a 4-tuple* $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ *where* $\mathcal{T}$ *is a TBox,* $\mathcal{P}$ *is a finite set of connected graph rules,* $\mathcal{G}$ *is a GBox, and* $\mathcal{A}$ *is an ABox.*

Next, we define the semantics of the formalism.

Definition 10 (Semantics)**.** *An interpretation* $I = (\triangle^I, \cdot^I)$ *is defined as usual, with the addition that it interprets each* ℓ*-ary description graph* G *as an* ℓ*-ary relation over* $\triangle^I$*—that is,* $G^I \subseteq (\triangle^I)^\ell$*. Each tuple in* G^I *is called a* graph instance *of* G*. A graph assertion is satisfied in* I*, written* $I \models G(a_1, \ldots, a_\ell)$*, if and only if* $\langle a_1^I, \ldots, a_\ell^I \rangle \in G^I$*. Satisfaction of a description graph, graph specialization, and graph alignment in* I *is defined in Table 4. A knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ *is satisfied in* I*, written* $I \models \mathcal{K}$*, if all its components are satisfied in* I*.*

The key and disjointness properties in Table 4 ensure that no two distinct instances of G can share a vertex; for example, different instances of G_{hand} cannot share the vertex that represents the thumb. These properties are required to ensure that the representation of the structured objects is bounded, which is a core

assumption of our formalism. In particular, they prevent the existence of infinite "chains" of instances of G_{hand}, which is crucial for the decidability of our formalism.

The start property in Table 4 ensures that each instance of a main concept A of G occurs in an instance of G. For example, since *Hand* is a main concept for G_{hand}, each instance of *Hand* must occur as vertex 1 in an instance of G_{hand}. Similarly, vertex 3 of G_{hand} is labeled with *Thumb*, which is the main concept of G_{thumb}; hence, each vertex 3 in an instance of G_{hand} is also a vertex 1 in an instance of G_{thumb} (but not the other way around). The disjunction in the start property handles the case when a main concept labels multiple vertices. For example, if we were to describe the hand and the five fingers in a single graph without a distinction between the five fingers, then, given an instance of a *Finger*, we would have to guess which of the five fingers we are dealing with. Finally, the layout property ensures that each instance of G is labeled and connected as specified in the definition of G.

Graph specializations are interpreted as inclusions over the graph relations; for example, axiom (22) states that each instance of a thumb is also an instance of a finger. The two graphs share all the vertices of the more general graph, and the more specific graph can introduce additional vertices.

Finally, graph alignments state that, whenever two graphs share some vertex from the specified list, then they share all other vertices from the list as well. For example, alignment (23) states that, if instances of G_{hand} and G_{thumb} share vertices 3 and 1, respectively, then they must also share vertices 4 and 3, respectively.

The main reasoning problem for graph-extended knowledge bases is satisfiability checking, since concept subsumption and instance checking can be reduced to satisfiability as usual.

Description graphs allow us to faithfully represent the nontree connections between the parts of a structured object. For example, consider the graph-extended knowledge base $\mathcal{K}'' = (\mathcal{T}'', \emptyset, \mathcal{G}'', \emptyset)$ where $\mathcal{T}''$ contains axioms (13)–(20), and $\mathcal{G}''$ contains description graph G_{index_finger} shown in Figure 2e. The GBox $\mathcal{G}''$ correctly axiomatizes the structure of the index finger and, unlike the DL knowledge base $\mathcal{K}'$ from Section 3, the graph-extended knowledge base $\mathcal{K}''$ entails axiom (21).

Note that Definition 10 does not ensure that objects in an instance of a description graph G are connected only by the edges as specified in G—that is, the maximum cardinality of the edges in an instance of G is not fixed. Because of that, axioms (13)–(19) are strictly necessary for the inference from the previous paragraph. Although Definition 10 could be straightforwardly extended with conditions that impose appropriate cardinality restrictions, we refrain from doing so for several reasons. First, cardinalities can always be axiomatized explicitly as shown in the previous example, so the presented formalism is more general. Second, the adopted approach allows us to study at a finer-grained level the impact of various constructs on the decidability of reasoning.

5 Undecidability of Reasoning

Checking the satisfiability of a graph-extended knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ is clearly undecidable since the combination of DLs such as $\mathcal{ALC}$ with unrestricted Horn rules is already undecidable [26]. We next show that, even if $\mathcal{T} = \emptyset$, checking the satisfiability of $\mathcal{K}$ is undecidable due to an interaction between description graphs and unrestricted Horn rules. In our proof, we use the well-known undecidable problem of checking emptiness of the intersection of context-free grammars [18]. A *context-free grammar* is a tuple $\mathbf{G} = (\mathbf{T}, \mathbf{N}, \mathbf{P}, S)$, where $\mathbf{T}$ is a finite set of terminal symbols, $\mathbf{N}$ a finite set of nonterminal symbols, $S \in \mathbf{N}$ is a start symbol, and $\mathbf{P} \subseteq \mathbf{N} \times (\mathbf{T} \cup \mathbf{N})^*$ is a finite set of productions. The language generated by $\mathbf{G}$ is denoted as $L(\mathbf{G})$. Given two context-free grammars $\mathbf{G} = (\mathbf{T}, \mathbf{N}, \mathbf{P}, S)$ and $\mathbf{G}' = (\mathbf{T}, \mathbf{N}', \mathbf{P}', S')$ over the same set of terminal symbols $\mathbf{T}$ and with $\mathbf{N} \cap \mathbf{N}' = \emptyset$, checking whether $L(\mathbf{G}) \cap L(\mathbf{G}') = \emptyset$ is undecidable [18].

Proposition 1. *Checking the satisfiability of a graph-extended knowledge base $\mathcal{K} = (\emptyset, \mathcal{P}, \mathcal{G}, \mathcal{A})$ where all rules in $\mathcal{P}$ are Horn and $\mathcal{G} = (\mathcal{G}_G, \emptyset, \emptyset)$ is undecidable.*

Proof. Let $\mathbf{G} = (\mathbf{T}, \mathbf{N}, \mathbf{P}, S)$ and $\mathbf{G}' = (\mathbf{T}, \mathbf{N}', \mathbf{P}', S')$ be two context-free grammars with $\mathbf{N} \cap \mathbf{N}' = \emptyset$. We first construct a graph-extended knowledge base $\mathcal{K}_{\mathbf{G},\mathbf{G}'}$ that simulates the derivations of $\mathbf{G}$ and $\mathbf{G}'$. Let R_P be an atomic role uniquely associated with each symbol $P \in \mathbf{T} \cup \mathbf{N} \cup \mathbf{N}'$. The knowledge base $\mathcal{K}_{\mathbf{G},\mathbf{G}'}$ contains a Horn rule of the form (24) for each production in $\mathbf{P} \cup \mathbf{P}'$ of the form $P \to Q_1.Q_2. \ldots .Q_n$.

$$R_{Q_1}(x_0, x_1) \wedge R_{Q_2}(x_1, x_2) \wedge \ldots \wedge R_{Q_n}(x_{n-1}, x_n) \to R_P(x_0, x_n) \tag{24}$$

For each terminal symbol $P \in \mathbf{T}$, the GBox of $\mathcal{K}_{\mathbf{G},\mathbf{G}'}$ contains description graphs $G_P^1 = (V_P^1, E_P^1, \lambda_P^1, M_P^1)$ and $G_P^2 = (V_P^2, E_P^2, \lambda_P^2, M_P^2)$ defined as follows:

$$G_P^1: \quad \begin{array}{rr} V_P^1 = \{1,2\} & M_P^1 = \{A\} \\ \lambda_P^1\langle 1\rangle = \{A\} & \lambda_P^1\langle 2\rangle = \{B\} \end{array} \quad 1 \xrightarrow{R_P} 2$$

$$G_P^2: \quad \begin{array}{rr} V_P^2 = \{1,2\} & M_P^2 = \{B\} \\ \lambda_P^2\langle 1\rangle = \{B\} & \lambda_P^2\langle 2\rangle = \{A\} \end{array} \quad 1 \xrightarrow{R_P} 2$$

Finally, the ABox of $\mathcal{K}_{\mathbf{G},\mathbf{G}'}$ contains the assertion $A(a)$.

It is straightforward to see that $\mathcal{K}_{\mathbf{G},\mathbf{G}'}$ is satisfiable. Let I be any interpretation obtained by intersecting all models of $\mathcal{K}_{\mathbf{G},\mathbf{G}'}$ that have the same interpretation domain $\triangle^I$, and let $w = P_1. \ldots .P_n \in \mathbf{T}^*$ be a finite word over the set of nonterminal symbols $\mathbf{T}$. Due to the GBox of $\mathcal{K}_{\mathbf{G},\mathbf{G}'}$, domain objects $\{\alpha_0, \ldots, \alpha_n\} \subseteq \triangle^I$ exists such that $\langle \alpha_{i-1}, \alpha_i \rangle \in R_{P_i}^I$ for each $1 \leq i \leq n$. It is straightforward to see that the rules (24) encode the derivations of $\mathbf{G}$ and $\mathbf{G}'$—that is, for each nonterminal symbol $Q \in \mathbf{N}$ (resp. $Q \in \mathbf{N}'$), we have $\langle \alpha_0, \alpha_n \rangle \in R_Q^I$ if and only if a derivation of Q from w exists in $\mathbf{G}$ (resp. $\mathbf{G}'$). Now let $\mathcal{K}_{\mathbf{G},\mathbf{G}'}^{int}$ be the graph-extended knowledge obtained by extending $\mathcal{K}_{\mathbf{G},\mathbf{G}'}$ with the Horn rule (25).

$$R_S(x, y) \wedge R_{S'}(x, y) \to \bot \tag{25}$$

Clearly, $\mathcal{K}^{int}_{\mathbf{G},\mathbf{G}'}$ is unsatisfiable if and only if $L(\mathbf{G}) \cap L(\mathbf{G}') \neq \emptyset$, which proves the claim of this proposition. □

One might try to ensure decidability by requiring that $\mathcal{P} = \emptyset$. As we show next, this is not sufficient: an interaction between description graphs in $\mathcal{G}$ and number restrictions in $\mathcal{T}$ is another source of undecidability. In our proof, we present reductions from the well-known domino tiling problem. A *domino system* is a triple $\mathbf{S} = (\mathbf{D}, \mathbf{H}, \mathbf{V})$ where $\mathbf{D} = \{D_1, \ldots, D_m\}$ is a set of tiles, and $\mathbf{H} \subseteq \mathbf{D} \times \mathbf{D}$ and $\mathbf{V} \subseteq \mathbf{D} \times \mathbf{D}$ are *horizontal* and *vertical compatibility* conditions, respectively. An $\mathbf{S}$-*tiling* is a function $\tau : \mathbb{N} \times \mathbb{N} \to \mathbf{D}$ such that $\langle \tau(i,j), \tau(i+1,j) \rangle \in \mathbf{H}$ and $\langle \tau(i,j), \tau(i,j+1) \rangle \in \mathbf{V}$.[6] It is well known that, given a domino system $\mathbf{S}$, checking whether an $\mathbf{S}$-tiling exists is undecidable [7].

Proposition 2. *Checking the satisfiability of a graph-extended knowledge base $\mathcal{K} = (\mathcal{T}, \emptyset, \mathcal{G}, \mathcal{A})$ with $\mathcal{T}$ a TBox in $\mathcal{ALCF}$ and $\mathcal{G} = (\mathcal{G}_G, \emptyset, \emptyset)$ is undecidable.*

Proof. We first present a graph-extended KB $\mathcal{K}_{grid}$ that implies the existence of an infinite grid. The TBox of $\mathcal{K}_{grid}$ contains the following $\mathcal{ALCF}$ axioms:

$$\top \sqsubseteq \leq 1\, H \tag{26}$$

$$\top \sqsubseteq \leq 1\, V \tag{27}$$

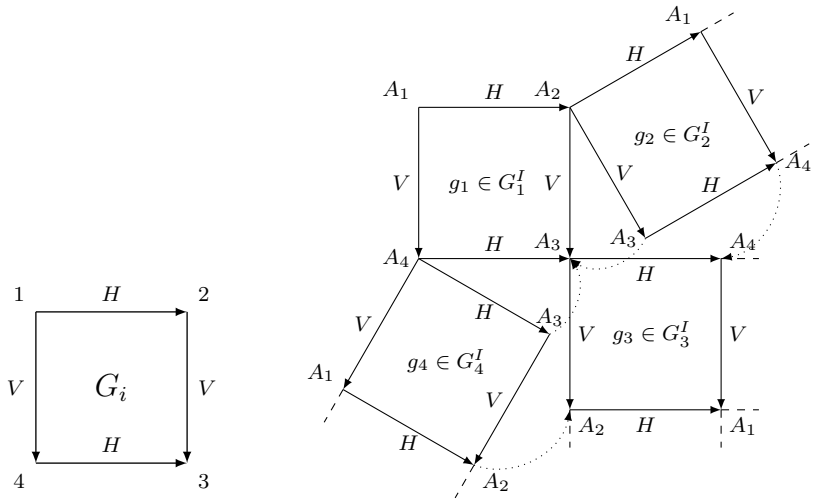
The ABox of $\mathcal{K}_{grid}$ consists of a single assertion $A_1(a)$. The GBox of $\mathcal{K}_{grid}$ contains four graphs $G_i = (V_i, E_i, \lambda_i, M_i)$, $1 \leq i \leq 4$. The sets of vertices V_i, edges E_i, and the labels of the edges are the same for all G_i and are shown in Figure 4a. Furthermore, $M_i = \{A_i\}$, and the labels λ_i of vertices in each G_i are as given next:

$$\begin{aligned}
\lambda_1 &= \{1 \mapsto \{A_1\}, 2 \mapsto \{A_2\}, 3 \mapsto \{A_3\}, 4 \mapsto \{A_4\}\} \\
\lambda_2 &= \{1 \mapsto \{A_2\}, 2 \mapsto \{A_1\}, 3 \mapsto \{A_4\}, 4 \mapsto \{A_3\}\} \\
\lambda_3 &= \{1 \mapsto \{A_3\}, 2 \mapsto \{A_4\}, 3 \mapsto \{A_1\}, 4 \mapsto \{A_2\}\} \\
\lambda_4 &= \{1 \mapsto \{A_4\}, 2 \mapsto \{A_3\}, 3 \mapsto \{A_2\}, 4 \mapsto \{A_1\}\}
\end{aligned}$$

We next show that $\mathcal{K}_{grid}$ is satisfiable and that each model I of $\mathcal{K}_{grid}$ contains an infinite two-dimensional grid, as shown in Figure 4b. Individual a corresponds to the top left corner of the grid, and instances of description graphs are labeled using lowercase letters. Since A_1 is a main concept for G_1, I contains the instance g_1 of G_1. Vertex 2 of g_1 is labeled with A_2, so I contains the instance g_2 of G_2; since V is functional by (27), the two graphs are "glued together" into a grid. By a similar argument, one can see that I contains the instance g_3 of G_3 and the instance g_4 of G_4, which are "glued" with g_1 and g_2 into a grid. By repeating the same argument, it is clear that the grid extends indefinitely to the right and downwards.

For a domino system $\mathbf{S} = (\mathbf{D}, \mathbf{H}, \mathbf{V})$, let $\mathcal{K}_{\mathbf{S}}$ be the graph-extended knowledge base obtained by extending the TBox of $\mathcal{K}_{grid}$ with the following axioms, where

[6] $\mathbb{N}$ is the set of all natural numbers.



(a) The GBox of $\mathcal{K}_{grid}$ (b) A Model I of $\mathcal{K}_{grid}$

Fig. 4. Grid Construction for $\mathcal{ALCF}$

each tile $D_i \in \mathbf{D}$, $1 \leq i \leq m$, corresponds to an atomic concept D_i:

$$\top \sqsubseteq D_1 \sqcup \ldots \sqcup D_m \tag{28}$$

$$D_i \sqcap D_j \sqsubseteq \bot \qquad \text{for each } 1 \leq i < j \leq m \tag{29}$$

$$D_i \sqcap \exists H.D_j \sqsubseteq \bot \qquad \text{for each } (D_i, D_j) \notin \mathbf{H} \tag{30}$$

$$D_i \sqcap \exists V.D_j \sqsubseteq \bot \qquad \text{for each } (D_i, D_j) \notin \mathbf{V} \tag{31}$$

These axioms ensure that each point in a grid is labeled with some D_i according to the compatibility conditions of $\mathbf{S}$. Thus, if $\mathcal{K}_\mathbf{S}$ is satisfiable, an $\mathbf{S}$-tiling can be extracted from a model of $\mathcal{K}_\mathbf{S}$; conversely, if an $\mathbf{S}$-tiling exists, a model of $\mathcal{K}_\mathbf{S}$ can be obtained by labeling vertices in the grid shown in Figure 4b according to the $\mathbf{S}$-tiling. These two facts then imply the claim of this proposition. □

The proofs of Propositions 1 and 2 suggest that undecidability arises partly because graph-extended knowledge bases can axiomatize models containing unbounded sequences of description graph instances. In practice, however, structured objects are usually modeled up to a certain level of granularity, which naturally determines a bound on the sequence of graphs one needs to represent. In such cases, we can describe the structure of an object by an acyclic hierarchy of parts; for example, carpal bones are parts of the hand, but the hand is not a part of the carpal bones. To reflect the acyclic nature of such a representation, it therefore seems reasonable to impose an acyclicity condition in our formalism. Intuitively, this condition ensures that the description graphs are arranged in a hierarchical manner and that their instantiation always provides a bounded representation.

Definition 11 (Acyclic GBox)**.** *Let $\mathcal{G} = (\mathcal{G}_G, \mathcal{G}_S, \mathcal{G}_A)$ be a GBox, and let $\lhd$ be the reflexive–transitive closure of the graph specialization relation $\lhd$ in $\mathcal{G}_S$. The GBox $\mathcal{G}$ is* acyclic *if a strict (i.e., an irreflexive and transitive, but not necessarily total)*

order $\prec$ on $\mathcal{G}_G$ exists such that, for each $G = (V, E, \lambda, M)$ and $G' = (V', E', \lambda', M')$ in $\mathcal{G}_G$ such that $G \not\preceq G'$, and for each $A \in M'$, the following two conditions hold:

- *$G' \lhd G$ implies $\neg A \in \lambda\langle i \rangle$ for each $i \in V \setminus V'$; and*
- *$G' \not\lhd G$ implies $\neg A \in \lambda\langle i \rangle$ for each $i \in V$.*

A graph-extended knowledge base is acyclic *if its GBox is acyclic.*

Intuitively, $G_1 \prec G_2$ means that an instance of G_1 is allowed to imply the existence of an instance of G_2. In our example, we would have $G_{hand} \prec G_{finger}$ and $G_{hand} \prec G_{thumb}$, which allows an instance of a hand to imply the existence of a finger and/or a thumb. We would also have $G_{finger} \prec G_{thumb}$, since finger is more general than thumb. The conditions in Definition 11 state that, if $G_1 \not\prec G_2$, then an instance of G_2 cannot imply the existence of an instance of G_1 because each node of G_2 must be labeled with a negation of each start concept of G_1. For example, since $G_{thumb} \not\prec G_{hand}$, no vertex in an instance of G_{thumb} should ever become labeled with a main concept of G_{hand}. Effectively, this prevents cyclic implications between instances of description graphs.

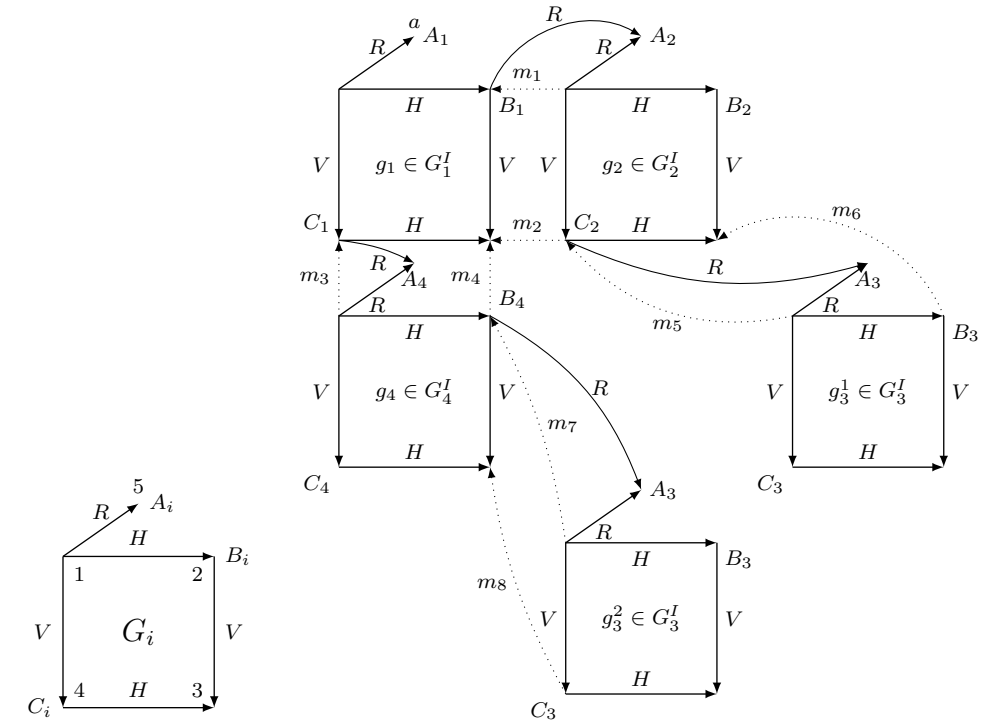
Requiring the GBox to be acyclic invalidates Proposition 1. In fact, checking the satisfiability of $\mathcal{K} = (\emptyset, \mathcal{P}, \mathcal{G}, \mathcal{A})$ where $\mathcal{G}$ is acyclic is decidable: $\mathcal{G}$ can then axiomatize only structures that can be obtained by unfolding $\mathcal{G}$ in a straightforward way, so it does not matter if the rules in $\mathcal{P}$ are not tree-like. Furthermore, in Section 6.2 we show that checking satisfiability of $\mathcal{K} = (\mathcal{T}, \emptyset, \mathcal{G}, \mathcal{A})$ is decidable if $\mathcal{T}$ is in $\mathcal{SHOQ}^+$. We next show, however, that an interaction between inverse roles, number restrictions, and description graphs leads to undecidability even if $\mathcal{G}$ is acyclic.

Proposition 3. *Checking the satisfiability of a graph-extended knowledge base $\mathcal{K} = (\mathcal{T}, \emptyset, \mathcal{G}, \mathcal{A})$ with $\mathcal{T}$ a TBox in $\mathcal{ALCIF}$ and $\mathcal{G} = (\mathcal{G}_G, \emptyset, \emptyset)$ an acyclic GBox is undecidable.*

Proof. Let $\mathcal{K}_{grid}$ be the graph-extended KB in which the GBox contains four description graphs $G_i = (V_i, E_i, \lambda_i, M_i)$, $1 \leq i \leq 4$ with the structure as shown in Figure 5a and $M_i = \{A_i\}$. To make the GBox acyclic, we assume that all vertices in each G_i are labeled with $\neg A_j$ for $i \neq j$; these negative labels are not shown in Figure 5a for the sake of clarity. The ABox of $\mathcal{K}_{grid}$ contains the assertion $A_1(a)$. Finally, the TBox of $\mathcal{K}_{grid}$ contains the following $\mathcal{ALCIF}$ axioms:

$$\begin{array}{lll} \top \sqsubseteq \leq 1\, H & \top \sqsubseteq \leq 1\, V & \top \sqsubseteq \leq 1\, R^- \\ B_1 \sqsubseteq \exists R.A_2 & C_1 \sqsubseteq \exists R.A_4 & \\ B_2 \sqsubseteq \exists R.A_1 & C_2 \sqsubseteq \exists R.A_3 & \\ B_3 \sqsubseteq \exists R.A_4 & C_3 \sqsubseteq \exists R.A_2 & \\ B_4 \sqsubseteq \exists R.A_3 & C_4 \sqsubseteq \exists R.A_1 & \end{array}$$

We next show that $\mathcal{K}_{grid}$ is satisfiable and that each model I of $\mathcal{K}_{grid}$ contains an infinite two-dimensional grid, as shown in Figure 5b. Instances of description graphs are denoted with lowercase letters. Due to the ABox assertion $A_1(a)$, I contains the instance g_1 of G_1. Due to $B_1 \sqsubseteq \exists R.A_2$, vertex 2 of g_1 is connected with an



(a) The GBox of $\mathcal{K}_{grid}$ (b) A Model I of $\mathcal{K}_{grid}$

Fig. 5. Grid Construction for $\mathcal{ALCIF}$

instance of A_2, so I contains the instance g_2 of G_2. Since R is inverse-functional, however, vertex 1 of g_2 must be the same as vertex 2 of g_1, as shown by the arrow m_1. Furthermore, since V is functional, vertex 4 of g_2 must be the same as vertex 3 of g_1, as shown by the arrow m_2. Hence, g_1 and g_2 form the top two squares of the grid. By a similar argument, one can see that I contains the instance g_4 of G_4, and that g_1 and g_4 share the vertices as shown by the arrows m_3 and m_4. Finally, I contains the instances g_3^1 and g_3^1 of G_3, which share vertices with g_2 and g_4 as shown my the arrows m_5–m_8. Thus, g_3^1 and g_3^2 share vertex 1, so by the key property they must be the one and the same instance. Consequently, g_1, g_2, $g_3^1 = g_3^2 = g_3$, and g_4 are connected in I in a grid-like manner. Note that no two instances g_i share vertex 5, so I satisfies the concept $\neg A_i$ occurring in the label of each vertex of G_j for $i \neq j$. By repeating the same argument, it is clear that the grid extends indefinitely to the right and downwards.

Now given a domino system $\mathbf{S} = (\mathbf{D}, \mathbf{H}, \mathbf{V})$, we can extend $\mathcal{K}_{grid}$ with axioms (28)–(31) to a knowledge base $\mathcal{K}_{\mathbf{S}}$ that is satisfiable if and only if an $\mathbf{S}$-tiling exists, which proves our claim. □

This result can be intuitively understood as follows. Let G_1 and G_2 be description graphs with start concepts A_1 and A_2, respectively; furthermore, let g_1 and g_2 be instances of G_1 and G_2, respectively. Then, inverse roles and number restrictions can merge g_1 and g_2 such that the vertex of g_1 labeled with A_1 is not contained in g_2 and, conversely, the vertex of g_2 labeled with A_2 is not contained in g_1. Therefore,

even if all vertices of G_1 and G_2 are labeled with $\neg A_2$ and $\neg A_1$, respectively, it is still possible to enforce the existence of infinite non-tree-like structures.

6 Reasoning with Graph-Extended Knowledge Bases

In this section, we present an algorithm for reasoning with a graph-extended knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$. In order to overcome the undecidability results from the previous section, in Section 6.1 we introduce *role separation* as a way of attaining decidability. Intuitively, role separation places restrictions on the usage of atomic roles in $\mathcal{T}$, $\mathcal{P}$, and $\mathcal{G}$ in order to limit the possible interaction between different types of axioms. In Section 6.2 we then present a hypertableau-based algorithm for checking the satisfiability of $\mathcal{K}$. When $\mathcal{K}$ is *weakly separated* (i.e., when the roles occurring in $\mathcal{P}$ do not occur in $\mathcal{T}$ and vice versa), our algorithm provides a decision procedure if $\mathcal{T}$ is in $\mathcal{SHOQ}^+$ and $\mathcal{G}$ is acyclic $\mathcal{G}$ and a semidecision procedure if $\mathcal{T}$ is in $\mathcal{SHOIQ}^+$. Finally, in Section 6.3 we show that the decidability of the latter case can be regained by requiring $\mathcal{K}$ to be *strongly separated* (i.e., that the roles occurring in $\mathcal{P}$ and $\mathcal{G}$ do not occur in $\mathcal{T}$ and vice versa).

6.1 Role Separation

Let $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ be a graph-extended KB. In Section 5, we proposed acyclicity of $\mathcal{G}$ as a way to limit the size of the structures whose existence is implied by $\mathcal{G}$. As expected, acyclicity invalidates the proof of Proposition 2. The proof of Proposition 3, however, reveals that an interaction between graphs, inverse roles, and functionality axioms can circumvent the desired effects of acyclicity. Thus, one way of ensuring decidability is to restrict the interaction between $\mathcal{T}$, $\mathcal{P}$, and $\mathcal{G}$ by placing restrictions on the usage of roles. The general approach is captured by the following definition, which is specialized in the following sections.

Definition 12 (Role Separation)**.** *A* role separation scheme $\Lambda = (N_{R_\mathcal{T}}, N_{R_\mathcal{P}}, N_{R_\mathcal{G}})$ *is a triple where all $N_{R_\mathcal{T}}$, $N_{R_\mathcal{P}}$, and $N_{R_\mathcal{G}}$ are (not necessarily disjoint) subsets of the set of atomic roles N_R. The roles in $N_{R_\mathcal{T}}$, $N_{R_\mathcal{P}}$, and $N_{R_\mathcal{G}}$ are called $\mathcal{T}$-, $\mathcal{P}$-, and $\mathcal{G}$-roles, respectively. A graph-extended knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ is* Λ-separated *if all roles occurring in $\mathcal{T}$, $\mathcal{P}$, and $\mathcal{G}$ are $\mathcal{T}$-, $\mathcal{P}$-, and $\mathcal{G}$-roles, respectively.*

A similar idea has been used to ensure decidability of the fusions of Abstract Description Systems (ADSs) [5]: the component ADSs can share concepts, but the interaction through roles is restricted to ensure decidability. The separation into $\mathcal{T}$-, $\mathcal{P}$- and $\mathcal{G}$-roles is similar in spirit. ADSs, however, lack standard first-order variables, so they cannot directly represent arbitrarily connected structures and rules. The latter could potentially be axiomatized using an ADS; however, such an encoding is likely to be complex and therefore not practicable. Furthermore, fusions of ADSs require a strict separation of roles, which rules out weakly separated knowledge bases (see Section 6.2).

Similarly to the standard hypertableau algorithm presented in Section 2.3, our algorithm first preprocesses $\mathcal{K}$ into a set of rules $\mathcal{R}$, a GBox $\mathcal{G}$, and an ABox $\mathcal{A}$. We next define a notion of Λ-admissibility for $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ that identifies the types of inputs our core algorithm can handle. This notion closely parallels Definition 12 and, as we discuss shortly, there is a straightforward relationship between Λ-separation for knowledge bases and Λ-admissibility for the algorithm inputs.

Definition 13 (Admissibility)**.** *Let $\mathcal{R}$ be a set of rules, $\mathcal{G}$ a GBox, $\mathcal{A}$ an ABox, and $\Lambda = (N_{R_\mathcal{T}}, N_{R_\mathcal{P}}, N_{R_\mathcal{G}})$ a role separation scheme. The triple $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is* Λ-admissible *if all roles in $\mathcal{G}$ are $\mathcal{G}$-roles, $\mathcal{A}$ is an input ABox, and $\mathcal{R}$ can be separated into disjoint subsets $\mathcal{R}_\mathcal{T}$ and $\mathcal{R}_\mathcal{P}$ of $\mathcal{T}$- and $\mathcal{P}$-*rules*, respectively, such that*

- *each rule $r \in \mathcal{R}_\mathcal{T}$ is an HT-rule and all roles in r are $\mathcal{T}$-roles, and*
- *each rule $r \in \mathcal{R}_\mathcal{P}$ is a connected graph rule and all roles in r are $\mathcal{P}$-roles.*

A Λ-admissible triple $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is simple *if all rules in $\mathcal{R}_\mathcal{T}$ are simple HT-rules, and $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is* acyclic *if $\mathcal{G}$ is acyclic.*

Let $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ be a graph-extended KB, $\Xi_\mathcal{T}(\mathcal{T})$ and $\mathcal{A}' = \mathcal{A} \cup \Xi_\mathcal{A}(\mathcal{T})$ the result of preprocessing $\mathcal{T}$ using the preprocessing transformation from [32, Section 4.1] (see Section 2.3 for a summary), and $\mathcal{R} = \mathcal{P} \cup \Xi_\mathcal{T}(\mathcal{T})$. By inspecting the transformation it is straightforward to see that, for each role separation scheme Λ, if $\mathcal{K}$ is Λ-separated, then $(\mathcal{R}, \mathcal{G}, \mathcal{A}')$ is Λ-admissible; if $\mathcal{T}$ is in $\mathcal{SHOQ}^+$, then $(\mathcal{R}, \mathcal{G}, \mathcal{A}')$ is simple; and if $\mathcal{G}$ is acyclic, then $(\mathcal{R}, \mathcal{G}, \mathcal{A}')$ is acyclic as well. Furthermore, if $\mathcal{K}$ does not contain transitivity axioms, it is trivial to see that $\mathcal{K}$ is equisatisfiable with $(\mathcal{R}, \mathcal{G}, \mathcal{A}')$. Finally, if Λ is such that $N_{R_\mathcal{T}} \cap N_{R_\mathcal{P}} = \emptyset$, then there is no interaction between $\mathcal{T}$ and $\mathcal{P}$, so transitivity axioms in $\mathcal{T}$ can be encoded into GCIs in the same way as in [32, Section 4.1] without affecting satisfiability. Therefore, we omit the details of the preprocessing phase for the sake of brevity and present an algorithm that decides the satisfiability of an admissible triple $(\mathcal{R}, \mathcal{G}, \mathcal{A})$.

6.2 Weakly Separated Knowledge Bases

We now define a notion of weak role separation.

Definition 14 (Weak Separation)**.** *A role separation scheme $(N_{R_\mathcal{T}}, N_{R_\mathcal{P}}, N_{R_\mathcal{G}})$ is* weak *if $N_{R_\mathcal{T}} \cap N_{R_\mathcal{P}} = \emptyset$. A graph-extended knowledge base $\mathcal{K}$ is* weakly separated *if a weak role separation scheme Λ exists such that $\mathcal{K}$ is Λ-separated. Similarly, for $\mathcal{R}$ a set of rules, $\mathcal{G}$ a GBox, and $\mathcal{A}$ an ABox, $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is* weakly admissible *if a weak role separation scheme Λ exists such that $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is Λ-admissible.*

Intuitively, weak separation prevents any interaction between $\mathcal{T}$ and $\mathcal{P}$ and thus avoids well-known sources of undecidability, such as the ones identified in [26]. From a modeling point of view, weak separation is interesting because it allows one to describe general knowledge using TBox axioms and then to specialize such knowledge using description graphs. For example, even if the general structure of a finger were described using DL axioms (e.g., this description might be a part of a general, coarse-grained knowledge base that does not use description graphs), one

could describe more specialized knowledge, such as the structure of an index finger, using graphs. One can thus choose the appropriate style of modeling for knowledge at different levels of granularity. The main limitation of weak separation is that one cannot use rules to express knowledge about roles used in DL axioms. Thus, weak separation does not impose any additional restrictions on graph-extended KBs beyond those that are already present in standard DL knowledge bases without description graphs. Note that DL-safe [33] need not satisfy the weak admissibility requirement; however, we do not consider such rules in this paper because of their limited applicability to schema reasoning.

We next present an algorithm that can be used to check the satisfiability of a weakly separated knowledge base $\mathcal{K}$. The formal definitions of the algorithm are rather intricate, so we first outline the main ideas by means of an example. Consider the following graph-extended knowledge base $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{P}_1, \mathcal{G}_1, \mathcal{A}_1)$:

$$\begin{array}{ll} \mathcal{T}_1 = \{\ C \sqsubseteq \exists R.A, & \mathcal{G}_1 \text{ contains the following description graph:} \\ \qquad\quad B \sqsubseteq \{b\} \quad \} & \\ \mathcal{P}_1 = \emptyset & \qquad\quad V = \{1,2,3\} \quad \lambda\langle 1\rangle = \{A\} \quad 1 \xrightarrow{S} 2 \\ \mathcal{A}_1 = \{\ C(a)\ \} & G\text{:} \quad M = \{A\} \qquad \lambda\langle 2\rangle = \{B\} \quad 2 \xrightarrow{T} 3 \\ & \qquad\qquad\quad\ \ \lambda\langle 3\rangle = \{C\} \quad 1 \xrightarrow{U} 3 \end{array} \tag{32}$$

The preprocessing of $\mathcal{T}_1$ produces the ABox $\Xi_{\mathcal{A}}(\mathcal{T}_1) = \{O_b(b)\}$ and the following set of rules $\Xi_{\mathcal{T}}(\mathcal{T}_1)$:

$$C(x) \rightarrow (\exists R.A)(x) \tag{33}$$

$$B(x) \wedge O_b(y_b) \rightarrow x \approx y_b \tag{34}$$

Let $\mathcal{R}_1 = \Xi_{\mathcal{T}}(\mathcal{T}_1)$ and $\mathcal{A}_1^1 = \Xi_{\mathcal{A}}(\mathcal{T}_1) \cup \mathcal{A}_1$. Clearly, $(\mathcal{R}_1, \mathcal{G}_1, \mathcal{A}_1^1)$ is weakly admissible: all rules in $\mathcal{R}_1$ are HT-rules and, since $\mathcal{P} = \emptyset$, we can consider all roles to be $\mathcal{T}$-roles.

By successively applying the derivation rules shown in Tables 3 and 5 to $\mathcal{R}_1$, $\mathcal{G}_1$, and $\mathcal{A}_1^1$, our algorithm tries to construct an ABox that represents a model of $(\mathcal{R}_1, \mathcal{G}_1, \mathcal{A}_1^1)$. The evolution of $\mathcal{A}_1^1$ is shown in Figure 6, where assertions derived by a single application of a derivation rule are separated by dotted lines. Note that the derivation rules from Table 5 closely follow the semantic conditions on description graphs given in Definition 10.

The *Hyp*-rule derives new assertions based on the contents of $\mathcal{R}$: if the body of some rule $r \in \mathcal{R}$ can be matched to assertions in an ABox, an assertion from the head of r is derived nondeterministically. Thus, from $C(a)$ and (33), the *Hyp*-rule derives the assertion $\exists R.A(a)$.

To satisfy this assertion, the $\geq$-rule introduces a fresh *tree successor* s_1 of a and it derives the assertions $R(a, s_1)$ and $A(s_1)$. To keep track of the successor relation, our algorithm represents individuals as finite strings of the form $\triangleright.\alpha_1.\ \ldots\ .\alpha_n$, where α_i are *symbols*, and $\triangleright$ is a special symbol that is used to make certain definitions simpler. Thus, the individual a actually corresponds to the string $\triangleright.\nu_a$ where ν_a is a *name symbol*; furthermore, s_1 corresponds to the individual $\triangleright.\nu_a.\tau$, where τ is a

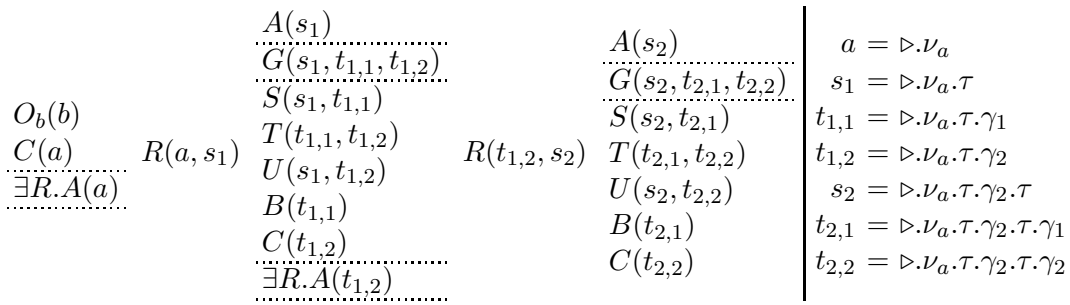


Fig. 6. Example Derivation of the Hypertableau Algorithm

tree symbol. That s_1 is a successor of a is evident from the fact that $a = \triangleright.\nu_a$ is a prefix of $s_1 = \triangleright.\nu_a.\tau$.

The concept A is a main concept in G so, due to the assertion $A(s_1)$, individual s_1 must occur in an instance of G at vertex 1; to ensure this, the hypertableau calculus contains the $G_\exists$-rule. An application of the $G_\exists$-rule to $A(s_1)$ derives the assertion $G(s_1, t_{1,1}, t_{1,2})$. Individuals $t_{1,1}$ and $t_{1,2}$ are fresh *graph successors* of s_1, which is reflected in their string representation: we have $t_{1,1} = s_1.\gamma_1 = \triangleright.\nu_a.\tau.\gamma_1$ and $t_{1,2} = s_1.\gamma_2 = \triangleright.\nu_a.\tau.\gamma_2$ where γ_1 and γ_2 are *graph symbols*. A tree or named individual and all of its graph successors are said to form a *cluster*; individuals s_1, $t_{1,1}$, and $t_{1,2}$ are an example of such a cluster.

In order to connect and label all the vertices in the instance of G, the hypertableau calculus contains the G_L-rule. Its application to the current set of assertions adds, among others, the assertion $C(t_{1,2})$. But then, the same inferences can be repeated: the *Hyp*-rule derives $\exists R.A(t_{1,2})$, the $\geq$-rule derives $R(t_{1,2}, s_2)$ and $A(s_2)$ where $s_2 = t_{1,2}.\tau = \triangleright.\nu_a.\tau.\gamma_2.\tau$, the $G_\exists$-rule derives the graph assertion $G(s_2, t_{2,1}, t_{2,2})$, and the G_L-rule connects and labels all the vertices. Let $\mathcal{A}_1^2$ be the ABox containing all assertions derived thus far; these are shown in Figure 6.

Clearly, unrestricted application of the $\geq$- and $G_\exists$-rule would lead to nontermination. Therefore, just like the standard (hyper)tableau algorithms, our algorithm applies *blocking*. Roughly speaking, tree individuals s_1 and s_2 occur in $\mathcal{A}_1^2$ in the same concepts, so the former individual blocks the latter—that is, the $\geq$- and $G_\exists$-rule are not applied to (the successors of) the blocked individual. Blocking is applicable because the ABox $\mathcal{A}_1^2$ is of structure that generalizes the notion of forest-shaped ABoxes from Section 2.3. In particular, $\mathcal{A}_1^2$ can thus be seen as consisting of three clusters, shown in Figure 6 as the left-most, middle, and right-most columns, connected by assertions $R(a, s_1)$ and $R(t_{1,2}, s_2)$.

In general, forest-shaped ABoxes are of the form shown in Figure 7. They contain several kinds of individuals, which we summarize next.

- *Root individuals* are shown in Figure 7 as black circles, and can be of two types:
  - *Named individuals* are the ones that occur in the input ABox.
  - Root individuals that are not named are introduced by the *NI*-rule (see Table 3) due to an interaction between inverse roles, number restrictions, and nominals.

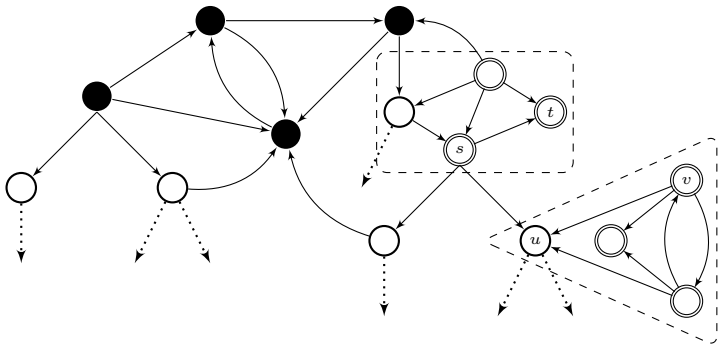


Fig. 7. A Generalized Forest-Shaped ABox

An in-depth discussion of the rationale behind the *NI*-rule is available in [32].

- *Tree individuals* are introduced by the $\geq$-rule in order to satisfy the existential quantifiers in the TBox of the knowledge base, and they are shown in Figure 7 as white circles with a single edge.
- *Graph individuals* are introduced by the $G_{\exists}$-rule in order to satisfy the start property for the graphs in the GBox of the knowledge base, and they are shown in Figure 7 as white circles with a double edge.

The central concept in forest-shaped ABoxes is the notion of a cluster, whose formal definition ensures that all root individuals and all graph individuals of the form $\triangleright.\gamma_i$ form a single cluster, and that each tree individual t and all graph individuals of the form $t.\gamma_i$ form a cluster. Figure 7 shows two clusters of two distinct tree individuals, where the member individuals are enclosed in a dashed line. The key idea behind clusters is that (i) individuals in the same cluster can be arbitrarily connected, but (ii) individuals from different clusters are connected in a tree-like manner. Thus, each forest-shaped ABox can be seen as a tree of clusters; we often call this structure a *tree backbone*. We exploit the tree backbone to generalize the notions of blocking and pruning from the standard (hyper)tableau algorithms.

In Lemma 1, we formalize the notion of forest-shaped ABoxes and show that, if $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is weakly separated, then the application of the hypertableau derivation rules to a forest-shaped ABox always produces a forest-shaped ABox. Intuitively, the arbitrarily shaped $\mathcal{P}$-rules in $\mathcal{R}$ can be applied only to assertions involving individuals in the same cluster, where they can introduce arbitrary connections; however, due to weak separation, they cannot affect the tree backbone. The tree backbone is constructed solely using the $\mathcal{T}$-rules in $\mathcal{R}$.

Nominals, however, introduce a slight complication. Consider again the ABox $\mathcal{A}_1^2$. From $B(t_{1,1})$, $O_b(b)$, and (34), the *Hyp*-rule derives $t_{1,1} \approx b$. The $\approx$-rule then *prunes* $t_{1,1}$ (i.e., it removes all graph and tree descendants or $t_{1,1}$) and replaces it with b; pruning is necessary in order to avoid nontermination due to repeated individual creation and merging, as in the so-called "yo-yo" problem [4]. After $t_{1,1}$ is replaced with b, the ABox contains the graph assertion $G(s_1, b, t_{1,2})$ in which b is not from the same cluster as s_1 and $t_{1,2}$; thus, the ABox is not forest shaped. This is remedied through *graph cleanup*: the mentioned assertion is replaced with $G(v_1, b, v_2)$,

where $v_1 = \triangleright.\gamma_1$ and $v_2 = \triangleright.\gamma_2$ are fresh graph individuals from the cluster of b. The cluster of s_1 and $t_{1,2}$ is thus merged into the cluster of b in order to make the resulting ABox forest shaped. Furthermore, if graph cleanup is subsequently applied to an assertion of the form $G(w_1, b, w_2)$, individuals w_1 and w_2 are replaced with v_1 and v_2, respectively. Reusing individuals in graph cleanup is sound because of the key property in Table 4, and it allows us to establish a bound on the number of individuals introduced by the cleanup.

We next define our algorithm formally. At this point, we assume the rules in $\mathcal{R}$ to be HT-rules, but do not assume them to be simple. Thus, the algorithm can be applied to a triple $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ obtained by preprocessing a graph-extended knowledge base whose TBox is in $\mathcal{SHOIQ}^+$.

Definition 15. *The hypertableau algorithm for checking the satisfiability of an admissible triple $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is obtained by modifying parts of Definition 2 as follows.*

Individuals. *Let Σ_τ, Σ_γ, and Σ_ν be countably infinite and mutually disjoint sets of* tree, graph, *and* name symbols, *respectively, none of which contains the special symbol $\triangleright$. The set Σ_ι of NI-*symbols *is the smallest set such that $\langle \alpha, R, B, i\rangle \in \Sigma_\iota$ for each $\alpha \in \Sigma_\gamma \cup \Sigma_\nu \cup \Sigma_\iota$, role R, literal concept B, and integer i.*

An individual *is a finite string of the form $\triangleright.\alpha_1. \ldots .\alpha_n$ with $n \geq 1$ such that*

- $\alpha_1 \in \Sigma_\nu \cup \Sigma_\gamma \cup \Sigma_\iota$,
- $\alpha_i \in \Sigma_\gamma \cup \Sigma_\tau$ *for* $2 \leq i \leq n$, *and*
- $\alpha_i \in \Sigma_\nu \cup \Sigma_\gamma \cup \Sigma_\iota$ *implies* $\alpha_{i+1} \notin \Sigma_\gamma$ *for* $1 \leq i \leq n$.

An individual with $\alpha_n \in \Sigma_\tau$ (resp. $\alpha_n \in \Sigma_\gamma$) is a tree *(resp.* graph*)* individual. *Furthermore, an individual of the form $\triangleright.\alpha$ is a* root *individual, and if $\alpha \in \Sigma_\nu$, the individual is* named. *Let N_A, N_I, and N_O be the sets of all individuals, all named individuals, and all root individuals, respectively.*

For each individual $x.\alpha \in N_A$ (with x possibly being equal to $\triangleright$), we say that $x.\alpha$ is a successor *of x, x is* predecessor *of $x.\alpha$, and* descendant *and* ancestor *are the transitive closures of successor and predecessor, respectively.*

Cluster. *For each individual $s \in N_A$, the function $\lfloor s \rfloor$ is defined as follows: $\lfloor s \rfloor = s$ if s is a tree individual; otherwise, $\lfloor s \rfloor = t$ for $s = t.\alpha$. Individuals s and t* are from the same cluster *if $\lfloor s \rfloor = \lfloor t \rfloor$.*

Graph Cleanup. *Let $\mathcal{A}$ be an ABox containing an assertion $G(u_1, \ldots, u_\ell)$ where some u_i and u_j are not from the same cluster, and $\lfloor u_i \rfloor$ is an ancestor of u_j. A* cleanup *of u_j is an ABox obtained from $\mathcal{A}$ by pruning u_j and then replacing in all the remaining assertions u_j with an individual t defined as follows:*

- *if $\mathcal{A}$ contains another graph assertion $G(v_1, \ldots, v_\ell)$ such that $u_i = v_i$ and v_j is from the same cluster as u_i, then $t = v_j$;*
- *otherwise, t is a fresh graph successor of $\lfloor u_i \rfloor$.*

A graph cleanup *of $\mathcal{A}$ is obtained from $\mathcal{A}$ by iteratively applying a cleanup to candidate individuals as long as possible and in any sequence that satisfies the following restriction: whenever cleanup is applicable to u_i and u_j such that u_i is an ancestor*

of u_j, cleanup is applied first to u_i.[7]

Merge Target. *An individual t is a* merge target *for an individual s if t is a named individual, or t is a root individual and s is not a named individual, or t is not a root individual and s is a descendant of $\lfloor t \rfloor$.*

Merging. *The ABox $\mathsf{merge}_{\mathcal{A}}(s \to t)$ is obtained from $\mathsf{prune}_{\mathcal{A}}(s)$ by replacing s with t in all assertions, and then applying a graph cleanup.*

Derivation Rules. *The derivation rules from Table 3 are extended with the ones from Table 5. In the NI-rule (see Table 3), for $u = \triangleright.\alpha$ a root individual, R a role, B a literal concept, and i an integer, $\mathsf{rootfor}(u, R, B, i) = \triangleright.\langle \alpha, R, B, i \rangle$.*

Rule Precedence. *The $\approx$-rule can be applied to a (possibly annotated) equality $s \approx t$ in an ABox $\mathcal{A}$ only if $\mathcal{A}$ does not contain an equality $s \approx t \,@^{u}_{\leq n\, R.B}$ to which the NI-rule is applicable. Furthermore, the $G_{\exists}$-rule is applicable to an ABox only if the $\bot$-, $\approx$-, $G_{\bot}$-, $G_{\approx}$-, $G_{\lhd}$-, and G_L-rule are not applicable to the ABox.*

If $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is weakly admissible and simple, the $\mathcal{T}$-rules in $\mathcal{R}$ are simple HT-rules so, as explained in Section 2.3, the *NI*-rule then never gets applied, no root individual occurring in a derivation involves an *NI*-symbol, and pairwise blocking can be simplified to single blocking. Therefore, we implicitly make these assumptions whenever $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is simple.

We next show that, if $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is weakly admissible, simple, and acyclic, the hypertableau algorithm becomes a decision procedure. Intuitively, if $\mathcal{R}$ is simple (i.e., if $\mathcal{T}$ is in $\mathcal{SHOQ}^{+}$), then different clusters cannot interact in an adverse way due to number restrictions. Consider, for example, the ABox shown in Figure 7. Individual u can be merged into t; however, t then "inherits" all main concepts asserted on u. Thus, if t and u occur in assertions with description graphs G_1 and G_2, respectively, such that $G_1 \not\leftarrow G_2$, an inconsistency will be derived due to the acyclicity of $\mathcal{G}$, which will prevent further application of the derivation rules. Hence, despite the fact that different clusters can be merged, we can establish a bound on the size of each cluster and thus prove termination.

We next prove soundness, completeness, and termination of our algorithm. To this end, we first formalize the intuitive notion of forest-shaped ABoxes and show that an application of a derivation rule always preserves this property.

Lemma 1. *Let $\mathcal{R}$ be a set of rules, $\mathcal{G}$ a GBox, and $\mathcal{A}$ an ABox such that $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is simple and weakly admissible. Then, each ABox $\mathcal{A}'$ labeling a node of a derivation for $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ satisfies the following properties, for a and b root individuals, $u_{(i)}$ individuals, $\gamma_i, \gamma_j \in \Sigma_\gamma$, and $\tau_i, \tau_j \in \Sigma_\tau$.*

(1) Each $R(s,t) \in \mathcal{A}'$ where R is a $\mathcal{T}$-role has the form $R(u, u.\tau_i)$, $R(u, a)$, or $R(u_1, u_2)$, where u_1 and u_2 are individuals from the same cluster.

(2) Each $s \approx t \in \mathcal{A}'$ is of the form $a \approx u$, $u_1 \approx u_2$, $u_1 \approx u_2.\tau_i$, or $u.\tau_i \approx u.\tau_j$, where u_1 and u_2 are individuals from the same cluster.

[7] Note that, due to the freedom in choice of t and the order in which cleanup is applied to candidate individuals, graph cleanup of $\mathcal{A}$ is not uniquely defined; however, for the purposes of our algorithm, *any* cleanup of $\mathcal{A}$ will suffice.

Table 5
Derivation Rules Related to Description Graphs

$G_{\approx}$-rule	If 1. $\{G(s_1,\ldots,s_\ell), G(t_1,\ldots,t_\ell)\} \subseteq \mathcal{A}$, 2. $s_i = t_i$ for some $1 \leq i \leq \ell$, 3. $\{s_j \approx t_j \mid 1 \leq j \leq \ell\} \not\subseteq \mathcal{A}$, and 4. neither s_i nor t_i is indirectly blocked for each $1 \leq i \leq \ell$ then $\mathcal{A}_1 := \mathcal{A} \cup \{s_j \approx t_j \mid 1 \leq j \leq \ell\}$.
$G_{\perp}$-rule	If 1. $\{G(s_1,\ldots,s_\ell), G(t_1,\ldots,t_\ell)\} \subseteq \mathcal{A}$, 2. $s_i = t_j$ for some $i \neq j$, and 3. neither s_i nor t_i is indirectly blocked for each $1 \leq i \leq \ell$ then $\mathcal{A}_1 := \mathcal{A} \cup \{\perp\}$.
$G_{\exists}$-rule	If 1. $A(s) \in \mathcal{A}$ such that $A \in M$ for some $G = (V, E, \lambda, M) \in \mathcal{G}_G$, 2. s is not blocked in $\mathcal{A}$, and 3. for each $v_i \in V_A = \{v_1,\ldots,v_n\}$, no individuals $u_1,\ldots,u_\ell$ exist such that $G(u_1,\ldots,u_\ell) \in \mathcal{A}$ and $u_{v_i} = s$ then $\mathcal{A}_i := \mathcal{A} \cup \{G(t_1,\ldots,t_\ell)\}$ for each $1 \leq i \leq n$ where $t_{v_i} = s$ and all other t_k are fresh graph successors of $\lfloor s \rfloor$.
G_L-rule	If 1. $G(s_1,\ldots,s_\ell) \in \mathcal{A}$ with $G = (V, E, \lambda, M)$, 2. $\{A(s_i) \mid A \in \lambda\langle i\rangle\} \cup \{R(s_i, s_j) \mid R \in \lambda\langle i, j\rangle\} \not\subseteq \mathcal{A}$, and 3. s_i is not indirectly blocked for each $1 \leq i \leq \ell$ then $\mathcal{A}_1 := \mathcal{A} \cup \{A(s_i) \mid A \in \lambda\langle i\rangle\} \cup \{R(s_i, s_j) \mid R \in \lambda\langle i, j\rangle\}$.
$G_{\lhd}$-rule	If 1. $G_1 \lhd G_2 \in \mathcal{G}_S$, 2. $G_2(s_1,\ldots,s_{\ell_2}) \in \mathcal{A}$, 3. $G_1(s_1,\ldots,s_{\ell_1}) \notin \mathcal{A}$, and 4. s_i is not indirectly blocked for each $1 \leq i \leq \ell$ then $\mathcal{A}_1 := \mathcal{A} \cup \{G_1(s_1,\ldots,s_{\ell_1})\}$.
$G_{\leftrightarrow}$-rule	If 1. $G_1[v_1,\ldots,v_n] \leftrightarrow G_2[w_1,\ldots w_n] \in \mathcal{G}_A$, 2. $\{G_1(s_1,\ldots,s_{\ell_1}), G_2(t_1,\ldots,t_{\ell_2})\} \subseteq \mathcal{A}$, 3. $s_{v_i} = t_{w_i}$ for some $1 \leq i \leq n$, 4. $\{s_{v_j} \approx t_{w_j} \mid 1 \leq j \leq n\} \not\subseteq \mathcal{A}$, and 5. neither s_i nor t_i is indirectly blocked for each $1 \leq i \leq \ell$ then $\mathcal{A}_1 := \mathcal{A} \cup \{s_{v_j} \approx t_{w_j} \mid 1 \leq j \leq n\}$.

(3) In each $G(u_1,\ldots u_\ell) \in \mathcal{A}'$ and $U(u_1,u_2) \in \mathcal{A}'$ with U a $\mathcal{P}$-role, u_i are all from the same cluster. Furthermore, for each graph individual u_0 in $\mathcal{A}'$, a tree or a root individual u_n from the same cluster as u_0 exists such that u_0 has a path to *u_n in $\mathcal{A}'$—that is, individuals $u_1,\ldots,u_{n-1}$ exist such that u_{i-1} and u_i occur together in a graph assertion in $\mathcal{A}'$ for each $1 \leq i \leq n$.*

(4) In each $O_a(u) \in \mathcal{A}'$ with O_a a nominal guard concept, the individual u is named. Furthermore, in each $\geq n\, R.B(u) \in \mathcal{A}'$, the concept B is not a nominal guard concept.

(5) For each tree individual t_n in $\mathcal{A}'$, individuals $s_0,\ldots,s_n$ and $t_0,\ldots,t_{n-1}$ exist such that (i) s_0 is a root individual, (ii) for each $1 \leq i \leq n$, individuals s_i and t_{i-1} are from the same cluster, and (iii) for each $0 \leq i \leq n$, individual t_i is a tree successor of s_i, and $R_i(s_i,t_i) \in \mathcal{A}'$ for some $\mathcal{T}$-role R_i.

Proof. Let $\Lambda = (N_{R_\mathcal{T}}, N_{R_\mathcal{P}}, N_{R_\mathcal{G}})$ be a weak role separation scheme and $\mathcal{R}_\mathcal{T}$ and $\mathcal{R}_\mathcal{P}$ the subsets of $\mathcal{R}$ satisfying the conditions of Definition 13. We prove this lemma by induction on the rule application. Since $\mathcal{A}$ is an input ABox, the induction base is trivial. Assume that the claim holds for an ABox $\mathcal{A}_0$ and consider the inferences deriving some ABox $\mathcal{A}_i$.

($\bot$- and $G_\bot$-rule) The ABox $\mathcal{A}_1$ trivially satisfies Conditions 1–5.

($G_\lhd$-, $G_\leftrightarrow$-, $G_\approx$-, and G_L-rule) These rules are always applied to individuals in the same cluster, so $\mathcal{A}_1$ satisfies Conditions 1–5.

($G_\exists$-rule) Assume that $\mathcal{A}_i$ is obtained by an application of the $G_\exists$-rule to an assertion $A(s) \in \mathcal{A}_0$. All individuals $t_1, \ldots, t_\ell$ introduced by the rule application are from the same cluster as s, so $\mathcal{A}_i$ satisfies Conditions 1–5.

($\geq$-rule) Assume that $\mathcal{A}_1$ is obtained by an application of the $\geq$-rule to an assertion $\geq n\, R.B(s) \in \mathcal{A}_0$. All individuals $t_1, \ldots, t_n$ introduced by the rule application are tree successors of s such that $R(s, t_i) \in \mathcal{A}_1$, and B is not a nominal guard concept, so $\mathcal{A}_1$ satisfies Conditions 1–5.

(*Hyp*-rule) Assume that $\mathcal{A}_i$ is obtained from $\mathcal{A}_0$ by an application of the *Hyp*-rule to a rule $r \in \mathcal{R}$. The rule r does not contain a nominal guard concept in the head, so $\mathcal{A}_i$ satisfies Condition 4. Furthermore, the *Hyp*-rule does not introduce fresh individuals, so Condition 5 trivially holds for $\mathcal{A}_i$.

If $r \in \mathcal{R}_\mathcal{P}$, then r is connected, so all variables in r are matched to individuals in the same cluster. All role atoms in the head of r are $\mathcal{P}$-roles due to weak separation, so $\mathcal{A}_i$ satisfies Conditions 1–3. Thus, let $r \in \mathcal{R}_\mathcal{T}$ be a simple HT-rule and consider the types of assertions derived by instantiating an atom from the head of r.

If $s \approx t$ is derived by instantiating an atom of the form $x \approx z_j$, then the body of r contains an atom $O_a(z_j)$. This atom is matched to an assertion $O_a(t) \in \mathcal{A}_0$ in which, by Condition 4, t is a named individual. Hence, $s \approx t$ satisfies Condition 2.

If $s \approx t$ is derived by instantiating $y_i \approx y_j$ in a simple HT-rule r, the body of r contains atoms $R(x, y_i)$ and $S(x, y_j)$ that are matched to assertions $R(u, s) \in \mathcal{A}_0$ and $S(u, t) \in \mathcal{A}_0$ where R and S are $\mathcal{T}$-roles, and each individual in $\{s, t\}$ is from the same cluster as u, or a tree successor of u, or a root individual. Clearly, $s \approx t$ satisfies Condition 2.

If $R(s, t)$ is derived by instantiating $R(x, x)$, then $s = t$; since s is from the same cluster as s, the assertion satisfies Condition 1.

If $R(s, t)$ is derived by instantiating $R(x, y_i)$ in a simple HT-rule r, the body of r contains an atom $S(x, y_i)$ that is matched to assertion $S(s, t) \in \mathcal{A}_0$, which satisfies Condition 1. Clearly, $R(s, t)$ then satisfies Condition 1 as well.

If $R(s, t)$ is derived by instantiating $R(x, z_j)$ in a simple HT-rule r, the body of r contains an atom $O_a(z_j)$ that is matched to an assertion $O_a(t) \in \mathcal{A}_0$. By Condition 4, t is a root individual, so $R(s, t)$ satisfies Condition 1.

($\approx$-rule) Consider the types of equality assertions in $\mathcal{A}_0$ to which the $\approx$-rule can be applied.

For $u_1 \approx u_2$ where u_1 and u_2 are from the same cluster, the $\approx$-rule prunes one individual—call it s—and replaces it with another individual from the same cluster. Clearly, the resulting assertions satisfy Conditions 1 and 2. Furthermore, individual

s occurs in the ABox after pruning only in assertions involving predecessors of s or individuals from the same cluster as s, so $\mathcal{A}_1$ satisfies Conditions 3–5 as well.

For $u_1 \approx u_2.\tau_i$ with u_1 and u_2 from the same cluster, the $\approx$-rule prunes $u_2.\tau_i$ and replaces it with u_1. Thus, assertions of the form $R(u_2, u_2.\tau_i)$, $R(u_2.\tau_i, a)$, $a \approx u_2.\tau_i$, $u_3 \approx u_2.\tau_i$ with u_2 and u_3 from the same cluster, and $u_2.\tau_i \approx u_2.\tau_j$ are changed into assertions $R(u_2, u_1)$, $R(u_1, a)$, $a \approx u_1$, $u_3 \approx u_1$, and $u_1 \approx u_2.\tau_j$, respectively, all of which satisfy Conditions 1 and 2. Furthermore, pruning removes all individuals from the cluster of $u_2.\tau_i$, so $\mathcal{A}_1$ satisfies Conditions 3–5 as well.

For $a \approx u$, the $\approx$-rule prunes u and merges it into a. Thus, assertions of the form $R(v, u)$ and $v \approx u$ are changed into assertions $R(v, a)$ and $v \approx a$, respectively, all of which satisfy Conditions 1 and 2. Replacing u with a in $G(u_1, \ldots, u_n)$ where $u_i = u$ produces at first an assertion that does not satisfy Condition 3; however, graph cleanup then replaces each u_j with a graph individual from the same cluster as a. Since $\mathcal{A}_0$ satisfies the second part of Condition 3, graph cleanup replaces all individuals from the cluster of u with graph individuals from the same cluster as a, so the resulting ABox satisfies Conditions 1–5. □

Theorem 1 summarizes the properties of our algorithm.

Theorem 1. *The following properties hold for each set of rules $\mathcal{R}$, GBox $\mathcal{G}$, and ABox $\mathcal{A}$ such that $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is weakly admissible, simple, and acyclic:*

(1) if $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is satisfiable, then each derivation for $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is successful;
(2) $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is satisfiable if a successful derivation for $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ exists; and
(3) each derivation for $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is finite.

Proof of Claim 1. The claim follows from the following property: if $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is satisfiable and $\langle \mathcal{A}_1, \ldots, \mathcal{A}_n \rangle$ is the result of applying a derivation rule to $\mathcal{R}$, $\mathcal{G}$, and $\mathcal{A}$, then $(\mathcal{R}, \mathcal{G}, \mathcal{A}_i)$ is satisfiable for some $1 \leq i \leq n$ (and, consequently, $\mathcal{A}_i$ is clash-free). The proof is the same as in [32, Lemma 13] for all but the $\approx$-rule, in which the application of graph cleanup is nonstandard. Let I be a model of $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ and consider an application of the $\approx$-rule to $s \approx t \in \mathcal{A}$, producing an ABox $\mathcal{A}_1$. Let $\mathcal{A}'$ be the ABox obtained from $\mathcal{A}$ by pruning s and then replacing it with t. Since $I \models s \approx t$, we have $s^I = t^I$, so clearly $I \models \mathcal{A}'$. The ABox $\mathcal{A}_1$ is obtained from $\mathcal{A}'$ by graph cleanup, which can additionally replace some individuals u_i with v_i. If v_i is fresh, we can extend I to obtain a model of $\mathcal{A}_1$; otherwise, by the definition of graph cleanup, $\mathcal{A}'$ contains graph assertions $G(\ldots, u_i, \ldots)$ and $G(\ldots, v_i, \ldots)$ so, by the key property from Definition 10, we have $u_j^I = v_j^I$ for each j. Clearly, $(\mathcal{R}, \mathcal{G}, \mathcal{A}_1)$ is satisfied in I. □

Proof of Claim 2. Let $\mathcal{A}'$ be a clash-free ABox labeling a leaf of a successful derivation for $(\mathcal{R}, \mathcal{G}, \mathcal{A})$. To prove the claim, we next show how to construct a model of $(\mathcal{R}, \mathcal{G}, \mathcal{A})$. To do this, we first introduce several definitions.

A *path* is a finite nonempty sequence of pairs of individuals $p = [\frac{x_0}{x'_0}, \ldots, \frac{x_n}{x'_n}]$. Let $\mathsf{tail}(p) = x_n$ and $\mathsf{tail}'(p) = x'_n$. Furthermore, let $q = [p \mid \frac{x_{n+1}}{x'_{n+1}}]$ denote the path

$[\frac{x_0}{x'_0}, \ldots, \frac{x_n}{x'_n}, \frac{x_{n+1}}{x'_{n+1}}]$; we say that q is a *successor* of p, and p is a *predecessor* of q. The set of all paths $\mathbb{P}(\mathcal{A}')$ is defined inductively as follows:

- $[\frac{a}{a}] \in \mathbb{P}(\mathcal{A}')$ if a is a root individual and it occurs in $\mathcal{A}'$;
- $[p \mid \frac{s'}{s'}] \in \mathbb{P}(\mathcal{A}')$ if $p \in \mathbb{P}(\mathcal{A}')$ and s' is a successor of $\mathsf{tail}(p)$, s' occurs in an assertion of $\mathcal{A}'$, and s' is not blocked in $\mathcal{A}'$; and
- $[p \mid \frac{s}{s'}] \in \mathbb{P}(\mathcal{A}')$ if $p \in \mathbb{P}(\mathcal{A}')$ and s' is a successor of $\mathsf{tail}(p)$, s' occurs in an assertion of $\mathcal{A}'$, and s' is directly blocked in $\mathcal{A}'$ by s.

Paths $p, q \in \mathbb{P}(\mathcal{A}')$ are *from the same cluster*, written $p \sim q$, if

- $p = q$, or
- $p = [\frac{a}{a}]$ and $q = [\frac{b}{b}]$ for a and b root individuals, or
- individuals $\mathsf{tail}(p)$ and $\mathsf{tail}(q)$ are from the same cluster and either p and q are successors of the same path or one path is a successor of the other path.

We now define an interpretation I as follows, for each atomic concept A, each $\mathcal{T}$-role R, each $\mathcal{P}$-role U, and each description graph G:[8]

$$
\begin{aligned}
\triangle^I &= \mathbb{P}(\mathcal{A}') \\
a^I &= [\tfrac{a}{a}] \text{ for each root individual } a \text{ that occurs in } \mathcal{A}' \\
a^I &= b^I \text{ if } a \neq b \text{ and } \|a\|_{\mathcal{A}'} = b \\
A^I &= \{p \mid A(\mathsf{tail}(p)) \in \mathcal{A}'\} \\
R^I &= \{\langle p, [\tfrac{a}{a}]\rangle \mid a \text{ is a root individual, } p \not\sim [\tfrac{a}{a}], \text{ and } R(\mathsf{tail}(p), a) \in \mathcal{A}'\} \cup \\
&\quad\ \{\langle p_1, p_2\rangle \mid p_1 \sim p_2 \text{ and } R(\mathsf{tail}(p_1), \mathsf{tail}(p_2)) \in \mathcal{A}'\} \cup \\
&\quad\ \{\langle p, [p \mid \tfrac{s}{s'}]\rangle \mid p \not\sim [p \mid \tfrac{s}{s'}] \text{ and } R(\mathsf{tail}(p), s') \in \mathcal{A}'\} \\
U^I &= \{\langle p_1, p_2\rangle \mid p_1 \sim p_2 \text{ and } U(\mathsf{tail}(p_1), \mathsf{tail}(p_2)) \in \mathcal{A}'\} \\
G^I &= \{\langle p_1, \ldots, p_\ell\rangle \mid p_i \sim p_j \text{ for } 1 \leq i < j \leq \ell \text{ and } G(\mathsf{tail}(p_1), \ldots, \mathsf{tail}(p_\ell)) \in \mathcal{A}'\}
\end{aligned}
$$

$\mathcal{A}'$ is an HT-ABox, so $\triangle^I$ is not empty. To prove $I \models (\mathcal{R}, \mathcal{G}, \mathcal{A})$, we first show that, for each $p_s = [q_s \mid \frac{s}{s'}]$ and each individual w, the following claims hold (*):

- *$R(s,s) \in \mathcal{A}'$ (resp. $A(s) \in \mathcal{A}'$) iff $\langle p_s, p_s\rangle \in R^I$ (resp. $p_s \in A^I$):* Immediate by the definition of I.
- *If $B(w) \in \mathcal{A}'$ and $\mathcal{L}_{\mathcal{A}'}(w) = \mathcal{L}_{\mathcal{A}'}(s')$ for B a literal concept, then $p_s \in B^I$:* The proof is immediate if B is atomic. If $B = \neg A$, since the $\bot$-rule is not applicable to $\mathcal{A}'$, we have $A(w) \notin \mathcal{A}'$; but then, $A(s') \notin \mathcal{A}'$ and $A(s) \notin \mathcal{A}'$, which by the previous case implies $p_s \notin A^I$.
- *If $\geq n\, R.B(s) \in \mathcal{A}'$, then $p_s \in (\geq n\, R.B)^I$:* By the definition of paths, s is not blocked. Since the $\geq$-rule is not applicable to $\geq n\, R.B(s)$, individuals $u_1, \ldots, u_n$ exist such that $R(s, u_i) \in \mathcal{A}'$ and $B(u_i) \in \mathcal{A}'$ for each $1 \leq i \leq n$, and $u_i \not\approx u_j \in \mathcal{A}'$ for each $1 \leq i < j \leq n$. Each assertion $R(s, u_i)$ satisfies Property (1) of HT-ABoxes, so each u_i can be of one of the following forms.

[8] The function $\|\cdot\|_{\mathcal{A}'}$ has been introduced in Definition 2.

- u_i is from the same cluster as s. If s is a tree individual, let $p_{u_i} = [p_s \mid \frac{u_i}{u_i}]$; otherwise, let $p_{u_i} = [q_s \mid \frac{u_i}{u_i}]$. By the definition of I and the facts that $R(s, u_i) \in \mathcal{A}'$ and $B(u_i) \in \mathcal{A}'$, we conclude $\langle p_s, p_{u_i} \rangle \in R^I$ and $p_{u_i} \in B^I$.
- u_i is a successor of s, but u_i and s are not from the same cluster. If u_i is directly blocked by v_i, let $p_{u_i} = [p_s \mid \frac{v_i}{u_i}]$; otherwise, let $p_{u_i} = [p_s \mid \frac{u_i}{u_i}]$. In either case, we have $R(\mathsf{tail}(p_s), u_i) \in \mathcal{A}'$, which, by the definition of I, implies $\langle p_s, p_{u_i} \rangle \in R^I$. Furthermore, $B(u_i) \in \mathcal{A}'$ and $\mathcal{L}_{\mathcal{A}'}(u_i) = \mathcal{L}_{\mathcal{A}'}(\mathsf{tail}(p_{u_i}))$ imply $p_{u_i} \in B^I$.
- u_i is a root individual. Let $p_{u_i} = [\frac{u_i}{u_i}]$. We have $R(s, u_i) \in \mathcal{A}'$ and $B(u_i) \in \mathcal{A}'$, which imply $\langle p_s, p_{u_i} \rangle \in R^I$ and $p_{u_i} \in B^I$.

Consider now each $1 \leq i < j \leq n$. In all cases, we have $\mathsf{tail}'(p_{u_i}) \not\approx \mathsf{tail}'(p_{u_j}) \in \mathcal{A}'$. Since $\bot \notin \mathcal{A}'$ and the $\bot$-rule is not applicable, we also have $\mathsf{tail}'(p_{u_i}) \neq \mathsf{tail}'(p_{u_j})$, which implies $p_{u_i} \neq p_{u_j}$. Thus, we conclude that $p_s \in (\geq n\, R.B)^I$.

For an assertion $\alpha' \in \mathcal{A}'$ of the form $a \approx b$ and $a \not\approx b$ with a and b named individuals, it is straightforward to see that $I \models \alpha'$. Furthermore, if α' is of the form $R(a, b)$ or $B(a)$, or $\geq n\, R.B(a)$ with a a named individual, (*) implies $I \models \alpha'$. Consider now each $\alpha \in \mathcal{A}$. If $\alpha \notin \mathcal{A}'$, then $\mathcal{A}'$ contains renamings that, when applied to α, produce an assertion $\alpha' \in \mathcal{A}'$; but then, $I \models \alpha$ by the definition of I.

We next show that $I \models \mathcal{R}$. Consider a simple HT-rule $r \in \mathcal{R}_\mathcal{T}$ with variables x, y_i, and z_j as in Definition 3, and a mapping μ of the variables to $\triangle^I$ such that $I, \mu \models B_m$ for each body atom B_m of r. Let σ be a mapping of the variables in r to individuals in $\mathcal{A}'$ defined as follows:

- $\sigma(x) = \mathsf{tail}(\mu(x))$;
- $\sigma(y_i) = \mathsf{tail}'(\mu(y_i))$ if $\mu(y_i)$ is a successor of $\mu(x)$ such that $\mu(y_i) \not\sim \mu(x)$;
- $\sigma(y_i) = \mathsf{tail}(\mu(y_i))$ in all cases not covered by the previous one; and
- $\sigma(z_j) = \mathsf{tail}(\mu(z_j))$.

Atom B_m can be of the form $A(x)$, $A(y_i)$, $A(z_j)$, $R(x, x)$, or $R(x, y_i)$, for R a $\mathcal{T}$-role. By the definition of paths, $\sigma(x)$ is not blocked. Furthermore, if $\mu(y_i)$ is a successor of $\mu(x)$, then $\sigma(y_i)$ is a successor of $\sigma(x)$; otherwise, $\sigma(y_i)$ is either from the same cluster as $\sigma(x)$ or it is a named individual. Finally, by the definition of I, we have $\sigma(B_m) \in \mathcal{A}'$. Each variable z_j occurs in r in an atom of the form $O_a(z_j)$; by Condition 4 of Lemma 1 and the definition of I, all paths in O_a^I are of the form $[\frac{b}{b}]$ for b a named individual, so $\mu(z_j)$ is of that form as well. The *Hyp*-rule is not applicable to r, $\mathcal{A}'$, and σ, so $\sigma(H_n) \in \mathcal{A}'$ for some head atom H_n of r. We have the following possibilities for the structure of H_n.

- Assume that H_n is of the form $C(x)$ for C a literal concept or a concept of the form $\geq n\, S.B$; thus, we have $C(\sigma(x)) \in \mathcal{A}'$. By (*), we then have $\mu(x) \in C^I$. Thus, $I, \mu \models r$.
- Assume that H_n is of the form $R(x, x)$; thus, we have $R(\sigma(x), \sigma(x)) \in \mathcal{A}'$. By (*), we then have $\langle \mu(x), \mu(x) \rangle \in R^I$. Thus, $I, \mu \models r$.
- Assume that H_n is of the form $B(y_i)$; thus, we have $B(\sigma(y_i)) \in \mathcal{A}'$. By the definition of blocking, we have $\mathcal{L}_{\mathcal{A}'}(\sigma(y_i)) = \mathcal{L}_{\mathcal{A}'}(\mathsf{tail}(\mu(y_i)))$; by (*), we then have

$p_{y_i} \in E_i^I$. Thus, $I, \mu \models r$.

- Assume that H_n is of the form $S(x, y_i)$, so $S(\sigma(x), \sigma(y_i)) \in \mathcal{A}'$. By the definition of I, we have $\langle p_x, p_{y_i} \rangle \in S_i^I$. Thus, $I, \mu \models r$.
- The case when H_n is of the form $S(x, z_j)$ is analogous to the previous one.
- Assume that H_n is of the form $x \approx z_j$; thus, we have $\sigma(x) \approx \sigma(z_j) \in \mathcal{A}$. Since the $\approx$-rule is not applicable to $\mathcal{A}'$, we have $\sigma(x) = \sigma(z_j)$. Since $\sigma(x)$ is a named individual, it cannot block other individuals, so $\mathsf{tail}'(\mu(x)) = \sigma(x)$, which implies $\mu(x) = \mu(z_j)$. Thus, $I, \mu \models r$.
- Assume that H_n is of the form $y_i \approx y_j$; thus, we have $\sigma(y_i) \approx \sigma(y_j) \in \mathcal{A}'$. Since the $\approx$-rule is not applicable to $\mathcal{A}'$, we have $\sigma(y_i) = \sigma(y_j)$. By Definition 3, the antecedent of r contains atoms $R(x, y_i)$ and $R(x, y_j)$; therefore, $\langle \mu(x), \mu(y_i) \rangle \in R^I$ and $\langle \mu(x), \mu(y_j) \rangle \in R^I$. By Condition 1 of Lemma 1 and the definition of I, path $\mu(y_i)$ can be either of the form $[\frac{a}{a}]$ for a a root individual, a successor of $\mu(x)$, or from the same cluster as $\mu(x)$; similar restrictions hold for $\mu(y_j)$. But then, $\sigma(y_i) = \sigma(y_j)$ implies $\mu(y_i) = \mu(y_j)$. Thus, $I, \mu \models r$.

Consider a rule $r \in \mathcal{R}_\mathcal{P}$ containing variables $x_1, \ldots, x_n$ and μ a mapping of these variables to $\triangle^I$ such that $I, \mu \models B_m$ for each body atom B_m of r. Since r is connected and each nonunary atom B_m involves either a $\mathcal{P}$-role or a description graph, Condition 3 of Lemma 1 and the definition of I imply that all paths $\mu(x_i)$ are from the same cluster. Let σ be a mapping of the variables in r to individuals in $\mathcal{A}'$ defined as $\sigma(x_i) = \mathsf{tail}(\mu(x_i))$ for $1 \leq i \leq n$. By the definition of I, we have $\sigma(B_m) \in \mathcal{A}'$ for each body atom B_m. Since the *Hyp*-rule is not applicable to r, $\mathcal{A}'$, and σ, then $\sigma(H_n) \in \mathcal{A}'$ for some head atom H_n. But then, by the definition of I, we have $I, \mu \models H_n$.

The proof that $I \models \mathcal{G}$ is completely analogous to the one in the previous paragraph and we omit it for the sake of brevity. □

Proof of Claim 3. We show that each derivation (T, ρ) for $\mathcal{R}$, $\mathcal{G}$, and $\mathcal{A}$ satisfies the following properties: (1) if a derivation rule is applied to a subset of $\rho(g)$ for some derivation node $g \in T$, then the same derivation rule cannot become applicable to the same assertions in $\rho(g')$ for some descendant node g' of g; (2) an integer $\wp$ depending only on $\mathcal{R}$, $\mathcal{G}$, and $\mathcal{A}$ exists such that, for each $\rho(g)$ and each tree individual s in $\rho(g)$, the number of tree ancestors of s is at most $\wp$; (3) for each $\rho(g)$ and each tree individual s in it, the number of graph successors of s is bounded; (4) on each derivation path, the number of graph individuals introduced by graph cleanup is bounded; and (5) the number of root graph individuals in each $\rho(g)$ is bounded. Together, all these items imply that (6) the number of individuals introduced on each derivation path is bounded. Items (1) and (6) imply that the number of applications of all derivation rules on each derivation path is bounded as well, which implies the claim of this lemma.

(1) This item holds in exactly the same way as in the case of the standard hypertableau algorithm [32, Lemma 15]: if a derivation rule is applied to a subset of assertions of $\rho(g)$ for some derivation node $g \in T$, then assertions are added to

$\rho(g)$ that prevent a reapplication of the same derivation rule to the same assertions in $\rho(g')$ for some descendant node g' of g. We omit the details for the sake of brevity.

(2) Let c be the number of atomic concepts occurring in $(\mathcal{R}, \mathcal{G}, \mathcal{A})$, and let *depth* of a tree individual s, written $\mathsf{dep}(s)$, be the number of its tree ancestors. For each $g \in T$, the ancestors of each tree individual in $\rho(g)$ are present in $\rho(g)$ by Condition 5 of Lemma 1. Thus, if a tree individual s has depth $\wp = 2^c + 1$, two ancestors with the same individual label exist in $\rho(g)$, so s is blocked in $\rho(g)$. The $\geq$-rule is not applicable to blocked individuals, so the $\geq$-rule is never applied to such s. Thus, for each tree individual s in $\rho(g)$, we have $\mathsf{dep}(s) \leq \wp$.

(3) To prove this item, we first show a useful property. Let $\prec$ be an order on the description graphs in $\mathcal{G}$ that satisfies the conditions of Definition 11. Furthermore, let us assume that the hypertableau algorithm is modified such that each individual s in $\mathcal{A}'$ is assigned a *label* $\omega(s)$, which is a (possibly empty) string of the form

$$G_1|^{v_1 \to v'_1}. \ldots .G_n|^{v_n \to v'_n} \tag{35}$$

where $n \geq 0$, G_i is a description graph, and v_i and v'_i are vertices in G_i. Individuals are labeled according to the following rules:

- For s a tree or named individual, or a fresh graph individual introduced by graph cleanup, $\omega(s)$ is the empty string.
- If an application of the $G_\exists$-rule to an assertion $A(s)$ introduces a graph assertion $G(t_1, \ldots, t_\ell)$ with $s = t_i$ for some $1 \leq i \leq \ell$, then $\omega(t_j) = \omega(s).G|^{i \to j}$ for each $1 \leq j \leq \ell$ and $j \neq i$.

By induction on the applications of the derivation rules, we show that the following properties (†) hold for each clash-free ABox $\mathcal{A}'$ labeling a node of (T, ρ):

(i) For each graph individual s in $\mathcal{A}'$ with $\omega(s)$ of the form (35),
  (a) $\mathcal{A}'$ contains an assertion $G_n(u_1, \ldots, u_{\ell_n})$ such that $s = u_{v'_n}$;
  (b) $G_1 \prec \ldots \prec G_{n-1}$; and
  (c) if $G_{n-1} \not\prec G_n$, then $\mathcal{A}'$ contains (not necessarily distinct) graph assertions $G_n(w_1, \ldots, w_{\ell_n})$ and $G_n(w'_1, \ldots, w'_{\ell_n})$ such that $w_i = w'_j$ for some $i \neq j$.

(ii) For each individual s in $\mathcal{A}'$ such that $\lfloor s \rfloor$ is a tree individual, $\mathcal{A}'$ does not contain an individual t from the same cluster as s such that $s \neq t$ and $\omega(s) = \omega(t)$.

Property (†) clearly holds for the input ABox $\mathcal{A}$, so let $\mathcal{A}'$ be an ABox satisfying (†) and consider all possible derivation rules.

The *Hyp*-, $\bot$-, $G_\approx$-, $G_\bot$-, G_L-, $G_\lhd$-, and $G_\leftrightarrow$-rule do not introduce fresh individuals and do not remove assertions from an ABox, so they cannot invalidate (†). Furthermore, the $\geq$-rule introduces tree individuals t_i where $\omega(t_i)$ is the empty string, so the resulting ABox clearly satisfies (†).

Assume that the $\approx$-rule is applied to an assertion $s \approx t \in \mathcal{A}'$ and that the individual s is merged into t. By Lemma 1, pruning always removes either all or no graph individuals from some cluster; therefore, the ABox after merging, but before possible graph cleanup, clearly satisfies Condition (i-c) and (ii). Furthermore, if

merging requires graph cleanup, then freshly introduced graph individuals clearly satisfy Conditions (i) and (ii).

Consider an application of the $G_\exists$-rule to an assertion $A(s) \in \mathcal{A}'$ and an ℓ'-ary description graph $G' = (V', E', \lambda', M')$ with $A \in M'$. Assume that s is labeled as follows, for $G_n = (V_n, E_n, \lambda_n, M_n)$:

$$\omega(s) = G_1|^{v_1 \to v'_1}. \ \ldots \ .G_n|^{v_n \to v'_n}$$

By the induction assumption, $\omega(s)$ satisfies Conditions (i) and (ii). Moreover, by the rule precedence, the $G_\bot$-, G_L-, and $G_\lhd$-, and $\bot$-rule are not applicable to $\mathcal{A}'$.

The $G_\bot$-rule is not applicable, so $\mathcal{A}'$ does not contain assertions $G_n(w_1, \ldots, w_{\ell_n})$ and $G_n(w'_1, \ldots, w'_{\ell_n})$ such that $w_i = w'_j$ for some $i \neq j$. Since $\omega(s)$ satisfies Condition (i-a), we conclude that $G_{n-1} \prec G_n$. Furthermore, since $\omega(s)$ satisfies Condition (i-b), we have that $G_i \prec G_{i+1}$ for each $1 \leq i \leq n-1$.

By Condition (i-a), $\mathcal{A}'$ contains an assertion $G_n(u_1, \ldots, u_{\ell_n})$ such that $s = u_{v'_n}$. Since the G_L-rule is not applicable, $B(s) \in \mathcal{A}'$ for each $B \in \lambda_n\langle v'_n \rangle$. Since $\mathcal{G}$ is acyclic, by Definition 11 the following properties (‡) hold for each description graph G'' in $\mathcal{G}$ such that $G_n \not\preceq G''$ and each main concept C of G'':

- If $G'' \not\trianglelefteq G_n$, or if $G'' \trianglelefteq G_n$ and v'_n is not a vertex of G'', then $\neg C(s) \in \mathcal{A}'$.
- If $G'' \trianglelefteq G_n$ and v'_n is a vertex of G'', since the $G_\lhd$-rule is not applicable, $\mathcal{A}'$ contains an assertion $G''(q_1, \ldots, q_{\ell''})$ such that $q_{v'_n} = s$.

Let $G'(t_1, \ldots, t_{\ell'})$ be a graph assertion introduced by the $G_\exists$-rule such that $t_v = s$, and consider each fresh graph individual t_i, labeled as follows:

$$\omega(t_i) = G_1|^{v_1 \to v'_1}. \ \ldots \ .G_n|^{v_n \to v'_n}.G'|^{v \to i}$$

We next show that $\omega(t_i)$ satisfies Conditions (i) and (ii). Condition (i-a) is obviously satisfied, and we have already established that Condition (i-b) is satisfied by the induction assumption on $\omega(s)$. To show that $\omega(t)$ satisfies Condition (i-c), we consider all possible relationships between G_n and G'.

- $G_n \prec G'$: Condition (i-c) is vacuously satisfied.
- $G_n = G'$: Precondition 3 of the $G_\exists$-rule ensures that $v \neq v'_n$, so graph assertions $G'(t_1, \ldots, t_{\ell'})$ and $G_n(u_1, \ldots, u_{\ell_n})$ satisfy Condition (i-c).
- $G_n \not\preceq G'$: Consider the relationship between G_n and G' in $\trianglelefteq$.
  - $G' \not\trianglelefteq G_n$, or $G' \trianglelefteq G_n$ and v is not a vertex of G_n: By case 1 of (‡), we have that $\neg A(s) \in \mathcal{A}'$; since the $\bot$-rule is not applicable to $\mathcal{A}'$, we have $\bot \in \mathcal{A}'$, which is a contradiction.
  - $G' \trianglelefteq G_n$ and v is not a vertex of G_n: By case 2 of (‡), then $\mathcal{A}'$ contains an assertion $G'(q_1, \ldots, q_{\ell''})$ such that $q_v = s$; but then the $G_\exists$-rule is not applicable to $\mathcal{A}'$ by precondition 3, which is a contradiction.

Finally, to have $\omega(t_i) = \omega(q)$ for some individual q from the cluster of t_i, the $G_\exists$-rule must be applied to the same assertion for the same graph twice. By (1), this is not possible; hence, $\omega(t_i)$ satisfies (ii). This completes the proof of (†).

We next use (†) to show (3). Let g be the number of graphs and a the maximum arity of a graph in $\mathcal{G}$. The ordering $\prec$ is acyclic, so $n \leq g+1$ in each label of the form (35). Each label can thus be understood as a tuple of $g+1$ triples (G, v, v') where G is a description graph or is empty, and v and v' are integers between 1 and a. There are most $((g+1) \cdot a \cdot a)^{g+1}$ different such labels, and this number is bounded by $\vartheta = 2^{(2a+1)\cdot(g+1)^2}$. But then, by (ii), the number of graph individuals in a cluster of a tree individual in $\mathcal{A}'$ is bounded by ϑ as well, which implies Item (3).

(4) Consider any graph or tree individual s in an ABox $\mathcal{A}'$ labeling a derivation node. Let $t_1, \ldots, t_n$ be a sequence of tree individuals such that t_1 is a successor of a root individual, $t_n = \lfloor s \rfloor$, and each t_k is the closest tree predecessor of t_{k+1}. Furthermore, let $\chi_s = S_1, \ldots, S_n$ be a sequence where each S_k is the maximal subset of $\mathcal{A}'$ in which all individuals are from the cluster of t_k. In the worst case, s can be merged into a named individual a, and an individual from each S_k can be merged into S_{k+1}. By the first condition of the definition of graph cleanup, however, fresh graph individuals can be introduced at most once for each χ_s unique up to the renaming of individuals. By Item (2), $n \leq \wp$, and, by Item (3), the size of each S_i is bounded; therefore, the number of sequences χ_s unique up to the renaming of individuals is also bounded. Consequently, the number of fresh graph individuals introduced in graph cleanup is bounded as well.

(5) Item (4), property (†), and the fact that $\prec$ is acyclic imply that the number of fresh root graph individuals introduced by the $G_\exists$-rule is bounded. The proof is analogous to the proof of Item (3) and is omitted for the sake of brevity.

(6) By (4) and (5), the total number of root individuals in an ABox is bounded. Furthermore, by (1), (2), and (3), the number of their descendants is bounded as well. Therefore, by (1), the total number of applications of derivation rules on each derivation path is bounded as well. □

Since preprocessing of the TBox does not affect satisfiability of a graph-extended knowledge base, we immediately have the following theorem.

Theorem 2. *Checking the satisfiability of a weakly separated acyclic graph-extended knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ where $\mathcal{T}$ is in $\mathcal{SHOQ}^+$ is decidable.*

We now consider the case when $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ is a weakly separated acyclic graph-extended knowledge base with $\mathcal{T}$ in $\mathcal{SHOIQ}^+$. The TBox $\mathcal{T}$ is preprocessed as usual, so let $\mathcal{R} = \Xi_{\mathcal{T}}(\mathcal{T}) \cup \mathcal{P}$ and $\mathcal{A}' = \mathcal{A} \cup \Xi_{\mathcal{A}}(\mathcal{A})$; then, $(\mathcal{R}, \mathcal{G}, \mathcal{A}')$ is weakly admissible and acyclic, but not simple. By Proposition 3, checking the satisfiability of $(\mathcal{R}, \mathcal{G}, \mathcal{A}')$ is undecidable; consequently, the hypertableau algorithm does not necessarily terminate. Consider again Figure 7. If $\mathcal{R}$ is not simple (i.e., if $\mathcal{T}$ contains inverse roles), then v can be merged into s. Individual v, however, does not need to occur in an assertion involving a main concept of some graph from the cluster of u. Thus, the algorithm does not necessarily derive a contradiction if a graph from the cluster of u is not subordinate to all graphs in the cluster of s. Hence, even though $\mathcal{G}$ is acyclic, clusters can be of arbitrary size which leads to nontermination.

Our algorithm, however, can be used as a semidecision procedure. This is a non-trivial and practically interesting consequence since the algorithm uses blocking. Assume that the $\mathcal{T}$-rules in $\mathcal{R}$ have been obtained from a cyclic TBox, and that $(\mathcal{R}, \mathcal{G}, \mathcal{A}')$ is satisfiable. Blocking "increases the chances" for termination; in fact, provided that $\mathcal{G}$ is acyclic, our algorithm will necessarily terminate unless it performs one particular type of inference. Our algorithm is a semidecision procedure even if $\mathcal{G}$ is not acyclic, but then it is unlikely to terminate on satisfiable $(\mathcal{R}, \mathcal{G}, \mathcal{A}')$.

Since the rules in $\mathcal{R}_\mathcal{T}$ are not simple, pairwise blocking must be used, and the *NI*-rule can become applicable. Furthermore, as usual in the case of semidecision procedures, derivations must be *fair*; intuitively, this means that no application of an inference rule should be "postponed" infinitely often.

Definition 16 (Fair Derivation)**.** *A derivation (T, ρ) for $\mathcal{R}$, $\mathcal{G}$, and $\mathcal{A}$ is* unfair *if a branch $t_1, t_2, \ldots$ of T exists such that, for infinitely many nodes $t_{i_1}, t_{i_2}, \ldots$ on that branch, the same derivation rule is applicable to the same assertions in each $\rho(t_{i_j})$.* Fair *is the opposite of unfair.*

The correctness proofs for the standard hypertableau algorithm for $\mathcal{SHOIQ}^+$ [32] are quite involved and lengthy, and so is their generalization to graph-extended knowledge bases. To keep this paper within reasonable length, we only sketch the proofs of our claims. The full proofs can be obtained by a rather straightforward combination of the proofs of Lemma 1 and Theorem 1 and the proofs from [32].

We first generalize Lemma 1 to take into account the assertions that can be derived when $\mathcal{R}$ is not simple.

Lemma 2. *Let $\mathcal{R}$ be a set of rules, $\mathcal{G}$ a GBox, and $\mathcal{A}$ an ABox such that $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is weakly admissible. Then, each ABox $\mathcal{A}'$ labeling a node of a derivation for $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ satisfies Conditions (3)–(5) of Lemma 1, as well as the following conditions, for a and b root individuals, $u_{(i)}$ individuals, $\gamma_i, \gamma_j \in \Sigma_\gamma$, and $\tau_i, \tau_j \in \Sigma_\tau$.*

(1) Each $R(s,t) \in \mathcal{A}'$ where R is a $\mathcal{T}$-role has the form $R(u, u.\tau_i)$, $R(u.\tau_i, u)$, $R(u, a)$, $R(a, u)$, or $R(u_1, u_2)$ with u_1 and u_2 from the same cluster.

(2) Each equality in $\mathcal{A}'$ either is of the form $s \approx t\, @^a_{\leq n\, R.B}$ with s a tree individual that is not a successor of a and t a tree individual, or it is a possibly annotated equality of the form $a \approx u$, $u_1 \approx u_2$, $u_1 \approx u_2.\tau_i$, $u \approx u.\tau_i.\gamma_j$, $u.\tau_i \approx u.\tau_j$, or $u \approx u.\tau_i.\tau_j$, where u_1 and u_2 are individuals from the same cluster.

Proof (Sketch). The proof is a straightforward combination of the proofs of Lemma 1 and [32, Lemma 12]. The main difference is in the application of the $\approx$-rule to and equality of the form $u \approx u.\tau_i.\gamma_j$, which prunes $u.\tau_i.\gamma_j$ and merges it into u. The cluster of $u.\tau_i.\gamma_j$ is not necessarily pruned as well; however, graph cleanup ensures that all individuals from the cluster of $u.\tau_i.\gamma_j$ are replaced with individuals from the cluster of u. □

Theorem 3. *The following properties hold for each set of rules $\mathcal{R}$, a GBox $\mathcal{G}$, and an ABox $\mathcal{A}$ such that $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is weakly admissible:*

(1) if $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ *is satisfiable, then each derivation for* $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ *is successful; and (2)* $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ *is satisfiable if a successful and fair derivation for* $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ *exists.*

Proof (Sketch). The proof of Claim 1 is the same as in Theorem 1, apart from the case for the *NI*-rule which is the same as in [32, Lemma 13]. For Claim 2, let (T, ρ) be a successful fair derivation for $(\mathcal{R}, \mathcal{G}, \mathcal{A})$. The main difference to the proof of Claim 2 of Theorem 1 is that (T, ρ) is not necessarily finite. Let $t_1, t_2, \ldots$ be the branch of T such that each $\rho(t_i)$ is clash-free, and let $\mathcal{A}' = \bigcup_i \bigcap_{j \geq i} \rho(t_j)$. Since (T, ρ) is fair, no derivation rule is applicable to $\mathcal{A}'$. We can construct the model of $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ in the same way as in Theorem 1, with the following difference in the treatment of $\mathcal{T}$-roles:

$$
\begin{aligned}
R^I = \{&\langle [\tfrac{a}{a}], p\rangle \mid a \text{ is a root individual, } p \not\sim [\tfrac{a}{a}], \text{ and } R(a, \mathsf{tail}(p)) \in \mathcal{A}'\} \cup \\
&\{\langle p, [\tfrac{a}{a}]\rangle \mid a \text{ is a root individual, } p \not\sim [\tfrac{a}{a}], \text{ and } R(\mathsf{tail}(p), a) \in \mathcal{A}'\} \cup \\
&\{\langle p_1, p_2\rangle \mid p_1 \sim p_2 \text{ and } R(\mathsf{tail}(p_1), \mathsf{tail}(p_2)) \in \mathcal{A}'\} \cup \\
&\{\langle p, [p \mid \tfrac{s}{s'}]\rangle \mid p \not\sim [p \mid \tfrac{s}{s'}] \text{ and } R(\mathsf{tail}(p), s') \in \mathcal{A}'\} \cup \\
&\{\langle [p \mid \tfrac{s}{s'}], p\rangle \mid p \not\sim [p \mid \tfrac{s}{s'}] \text{ and } R(s', \mathsf{tail}(p)) \in \mathcal{A}'\}
\end{aligned}
$$

The proof that I is a model of $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is a straightforward combination of the proof of Claim 2 of Theorem 1 and [32, Lemma 14]. □

Termination is lost because the application of the $\approx$-rule to assertions of the form $u \approx u.\tau_i.\gamma_j$ invalidates Condition (ii) of (†) in the proof of Claim 3 of Theorem 1. The results in Section 6.3, however, show that the algorithm terminates if such an inference is not performed. A practical implementation can detect such inferences and warn the user about the loss of termination guarantees.

6.3 Strongly Separated Knowledge Bases

For $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ where $\mathcal{T}$ is in $\mathcal{SHOIQ}^+$, termination can be regained if $\mathcal{K}$ is strongly separated and acyclic. Then, for $\mathcal{R} = \Xi_{\mathcal{T}}(\mathcal{T}) \cup \mathcal{P}$ and $\mathcal{A}' = \mathcal{A} \cup \Xi_{\mathcal{A}}(\mathcal{A})$, triple $(\mathcal{R}, \mathcal{G}, \mathcal{A}')$ is strongly admissible and acyclic, as defined next.

Definition 17 (Strong Separation)**.** *A role separation scheme* $(N_{R_\mathcal{T}}, N_{R_\mathcal{P}}, N_{R_\mathcal{G}})$ *is* strong *if* $N_{R_\mathcal{T}} \cap N_{R_\mathcal{P}} = \emptyset$ *and* $N_{R_\mathcal{G}} = N_{R_\mathcal{P}}$*. A graph-extended knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ *is* strongly separated *if a strong role separation scheme* Λ *exists such that* $\mathcal{K}$ *is* Λ*-separated. Similarly, a triple* $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ *is* strongly admissible *if a strong role separation scheme* Λ *exists such that* $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ *is* Λ*-admissible.*

Strong separation restricts the modeling style in a more significant way than weak separation: essentially, it requires the modeler to determine in advance which knowledge will be modeled using DLs and which using graphs. Thus, knowledge modeled using DLs cannot be specialized using graphs and vice versa.

To understand why strong separation ensures decidability of reasoning, consider again Figure 7. The tree backbone then contains only $\mathcal{T}$-roles, and the clusters contain only $\mathcal{P}$-roles (i.e., $\mathcal{G}$-roles). Thus, the $\mathcal{T}$-rules from $\mathcal{R}$ can be applied only

to the tree backbone, while the $\mathcal{P}$-rules can be applied only to the graphs in a single cluster. Therefore, even if $\mathcal{R}$ is not simple, no rule in $\mathcal{T}$ can derive an equality that equates s with v and thus merges two distinct clusters.

Theorem 4. *Let $\mathcal{R}$ be a set of rules, $\mathcal{G}$ a GBox, and $\mathcal{A}$ an ABox such that $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is strongly admissible and acyclic. Then, each derivation for $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is finite.*

Proof (Sketch). Since $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ is strongly admissible, the rules in $\mathcal{R}$ can be separated into sets $\mathcal{R}_\mathcal{T}$ and $\mathcal{R}_\mathcal{P}$ of $\mathcal{T}$-rules and $\mathcal{P}$-rules, respectively, that do not share roles. By a straightforward modification to the proof of Lemma 2 one can see that, due to strong role separation, role assertions of the form $R(u_1, u_2)$, where R is a $\mathcal{T}$-role and u_1 and u_2 are from the same cluster are always of form $R(u, u)$ for u a tree individual. This can be used to strengthen Lemma 2 and show that each ABox $\mathcal{A}'$ labeling a node in a derivation for $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ contains equalities of the form $a \approx u$, $u_1 \approx u_2$, $u \approx u.\tau_i$, $u.\tau_i \approx u.\tau_j$, or $u \approx u.\tau_i.\tau_j$—that is, equalities of the form $u \approx u.\tau_i.\gamma_j$ are never derived.[9] As we show next, this can be used to show that the number of individuals introduced on a derivation path is bounded.

Assume that the $\approx$-rule is applied to $u \approx u.\tau_i$, $u.\tau_i \approx u.\tau_j$, or $u \approx u.\tau_i.\tau_j$. Just like in the proof of Claim 3 of Theorem 1, the merged individual is a tree individual, so its entire cluster is pruned and the resulting ABox satisfies (†); consequently, the number of individuals in the cluster of a tree individual is bounded by ϑ.

If the $\approx$-rule is applied to $a \approx u$, then the resulting ABox is subjected to graph cleanup; however, the number of newly introduced individuals is bounded in exactly the same way as in Item (4) of the proof of Claim 3 of Theorem 1.

It remains to be shown that the number of new root individuals introduced by the *NI*-rule is bounded as well. Let $\wp$ be the maximal number of tree ancestors of an individual occurring in $\mathcal{A}'$; in [32, Lemma 15], it was shown that $\wp$ is exponential in the number of atomic concepts and roles. Furthermore, in [32, Lemma 15] it was also shown that the root individuals introduced by the *NI*-rule can be seen as forming a tree with a polynomial branching factor and depth at most $\wp$. Since the number of graph individuals in each cluster is bounded, the addition of description graphs does not change the essence of this argument: the root individuals introduced by the *NI*-rule can be seen as forming a tree of clusters, where the size of each cluster is at most ϑ and the depth of the tree is at most $\wp$. Thus, the number of root individuals is bounded, which implies the claim of this theorem. □

Theorem 5. *Checking the satisfiability of a strongly separated acyclic graph-extended knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ where $\mathcal{T}$ is in $\mathcal{SHOIQ}^+$ is decidable.*

[9] Due to strong admissibility, equalities of the form $u_1 \approx u_2.\tau_i$ from Lemma 2 become $u \approx u.\tau_i$; however, this is not relevant to this termination proof.

7 Complexity of Reasoning

We now determine the exact complexity bounds of checking the satisfiability of a graph-extended knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$. In Section 7.1, we show that the problem is NExpTime-hard even if $\mathcal{T} = \emptyset$. Then, in Section 7.2, we show that the problem is in NExpTime if $\mathcal{K}$ is acyclic and weakly separated with $\mathcal{T}$ in $\mathcal{SHOQ}^+$, or if $\mathcal{K}$ is acyclic and strongly separated with $\mathcal{T}$ in $\mathcal{SHIQ}^+$. The case when $\mathcal{K}$ is strongly separated and acyclic and $\mathcal{T}$ is in $\mathcal{SHOIQ}^+$ is left for our future work.

7.1 Lower Bound

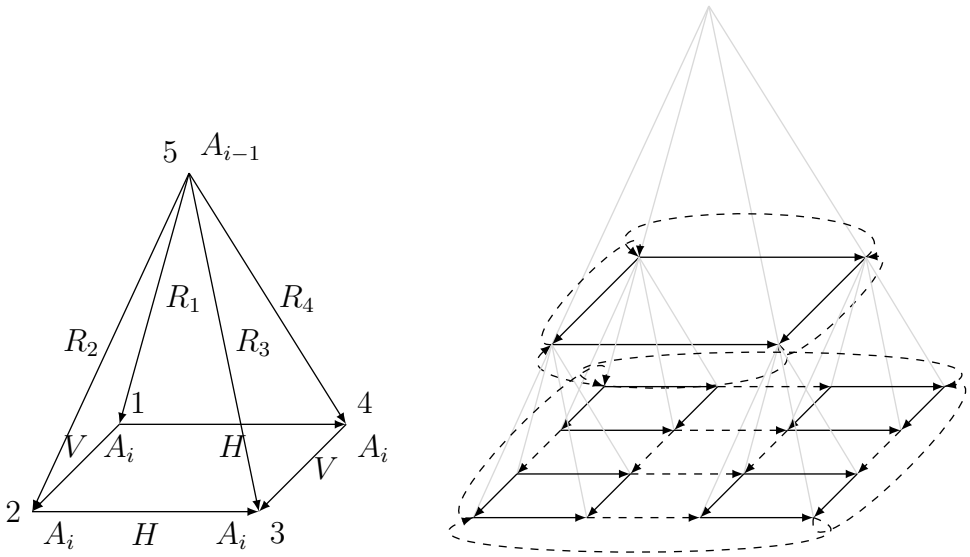
A graph-extended knowledge base $\mathcal{K}$ contains a set $\mathcal{P}$ or disjunctive datalog rules, and checking the satisfiability of $\mathcal{P}$ is NExpTime-complete [11] (under standard first-order semantics), so one might intuitively expect this result to provide a lower bound for the complexity of checking the satisfiability of $\mathcal{K}$. To understand why this is not the case, consider the following intuitive explanation of the result from [11]. The satisfiability of $\mathcal{P}$ alone can be decided by the following three-step process:

(1) Compute the grounding $\mathcal{P}_g$ of $\mathcal{P}$—that is, replace in $\mathcal{P}$ all variables in the rules with all individuals in all possible ways.
(2) Nondeterministically guess an interpretation I for $\mathcal{P}_g$.
(3) Check whether I is a model of $\mathcal{P}_g$.

Without restricting $\mathcal{P}$ in any way, the first and the third step can be implemented in exponential time, but the second step requires nondeterministic exponential time; thus, the overall complexity of this procedure is NExpTime. If, however, the arity of the predicates occurring in $\mathcal{P}$ is bounded, then the number of ground atoms in $\mathcal{P}_g$ is polynomial in $|\mathcal{P}|$, so all interpretations I can be enumerated by an exponential algorithm. Similarly, if the number of variables in $\mathcal{P}$ is bounded, then $\mathcal{P}_g$ is polynomial in $|\mathcal{P}|$; furthermore, in the second step we can clearly restrict our attention to interpretations that contain only the ground atoms from $\mathcal{P}_g$, so we can again enumerate all relevant interpretations in exponential time. Thus, for the problem to be NExpTime-hard, $\mathcal{P}$ must be allowed to contain predicates of arbitrary arity as well as rules with an arbitrary number of variables.

The set of rules $\mathcal{P}$ of a graph-extended knowledge base $\mathcal{K}$ can contain rules with an unbounded number of variables, and it can contain graph atoms with arbitrary arity. Graph atoms, however, must satisfy the disjointness and key properties from Definition 10; this imposes restrictions on the interpretation of graph atoms in addition to $\mathcal{P}$, so the hardness result from [11] does not apply. In fact, we show that checking the satisfiability of $\mathcal{K}$ is NExpTime-hard even if the rules are allowed to contain only unary and binary predicates and at least four variables. We thus identify a new source of complexity of reasoning with graph-extended knowledge bases: description graphs can succinctly encode exponential structures.

We prove hardness by a reduction from the *bounded domino tiling* problem [7]. Given a domino system $\mathbf{S} = (\mathbf{D}, \mathbf{H}, \mathbf{V})$, an *initial condition* for $\mathbf{S}$ is an n-



(a) The Structure of Each G_i (b) First Two Levels of the Pyramid

Fig. 8. Construction of an Exponential Grid

tuple $\mathbf{I} = D_{\alpha_0}, \ldots, D_{\alpha_{n-1}}$ of tiles from $\mathbf{D}$. A *bounded* $\mathbf{S}$*-tiling* for $\mathbf{I}$ is a function $\tau : \mathbb{N}_{2^n} \times \mathbb{N}_{2^n} \to \mathbf{D}$ where $\mathbb{N}_{2^n} = \{0, \ldots, 2^n - 1\}$, $\tau(i, 0) = D_{\alpha_i}$ for $0 \leq i < n$, and $\langle \tau(i, j), \tau(i \oplus_{2^n} 1, j) \rangle \in \mathbf{H}$ and $\langle \tau(i, j), \tau(i, j \oplus_{2^n} 1) \rangle \in \mathbf{V}$ where $\oplus_{2^n}$ denotes addition modulo 2^n. Given a domino system $\mathbf{S}$ and an initial condition $\mathbf{I}$, checking whether a bounded $\mathbf{S}$-tiling for $\mathbf{I}$ exists is NExpTime-complete [7].

Lemma 3. *Let $\mathcal{K} = (\emptyset, \mathcal{P}, \mathcal{G}, \mathcal{A})$ be a graph-extended KB where $\mathcal{G} = (\mathcal{G}_G, \emptyset, \emptyset)$ is an acyclic GBox and each rule in $\mathcal{P}$ contains only atomic concepts and roles and at most four variables. Then, checking the satisfiability of $\mathcal{K}$ is* NExpTime*-hard.*

Proof. For an arbitrary integer n, we first construct a graph-extended knowledge base $\mathcal{K}^n_{grid}$ that implies the existence of a "cyclic" grid with $2^n \times 2^n$ elements. The ABox of $\mathcal{K}^n_{grid}$ contains a single assertion $A_0(a)$. The GBox of $\mathcal{K}^n_{grid}$ contains n graphs $G_i = (V_i, E_i, \lambda_i, M_i)$, $1 \leq i \leq n$, as shown in Figure 8a, where $M_i = \{A_{i-1}\}$. The knowledge base $\mathcal{K}^n_{grid}$ contains the following rules:

$$A_0(x) \wedge R_4(x, y) \wedge R_1(x, z) \rightarrow H(y, z) \tag{36}$$
$$A_0(x) \wedge R_3(x, y) \wedge R_2(x, z) \rightarrow H(y, z) \tag{37}$$
$$A_0(x) \wedge R_2(x, y) \wedge R_1(x, z) \rightarrow V(y, z) \tag{38}$$
$$A_0(x) \wedge R_3(x, y) \wedge R_4(x, z) \rightarrow V(y, z) \tag{39}$$
$$H(x, y) \wedge R_4(x, z) \wedge R_1(y, w) \rightarrow H(z, w) \tag{40}$$
$$H(x, y) \wedge R_3(x, z) \wedge R_2(y, w) \rightarrow H(z, w) \tag{41}$$
$$V(x, y) \wedge R_2(x, z) \wedge R_1(y, w) \rightarrow V(z, w) \tag{42}$$
$$V(x, y) \wedge R_3(x, z) \wedge R_4(y, w) \rightarrow V(z, w) \tag{43}$$

We next show that $\mathcal{K}^n_{grid}$ is satisfiable and that each model I of $\mathcal{K}^n_{grid}$ contains a structure shown in Figure 8b. The individual a corresponds to the apex of the pyramid. Due to the assertion $A_0(a)$, the model I contains an instance of G_1 such that its vertex 5 corresponds to a. Vertices 1–4 of G_1 are labeled with A_1, so I contains four instances of G_2 as shown in the figure. By repeating this argument, I can be seen as containing a pyramid consisting of n levels, where level i contains $2^i \cdot 2^i$ vertices. Furthermore, in the first level, rules (36)–(39) ensure that vertex 4 is connected through H to vertex 1, vertex 3 is connected through H to vertex 2, vertex 2 is connected through V to vertex 1, and vertex 3 is connected through V to vertex 4; that is, the grid in the first level is "cyclic." Rules (40)–(43) then inductively use the H- and V-edges at level $i-1$ to construct the missing H- and V-edges at level i; since the grid at level $i-1$ is "cyclic," these rules construct at level i a "cyclic" grid as well. Thus, I contains at level n a "cyclic" grid of size $2^n \cdot 2^n$ in which all elements are labeled with A_n.

Consider now any domino system $\mathbf{S} = (\mathbf{D}, \mathbf{H}, \mathbf{V})$ with m tiles in $\mathbf{D}$ and any initial condition $\mathbf{I} = D_{\alpha_0}, \ldots, D_{\alpha_{n-1}}$. Let $\mathcal{K}_{\mathbf{S},\mathbf{I}}$ be a graph-extended knowledge base obtained by extending $\mathcal{K}^n_{grid}$ with the following rules, where each domino tile $D_i \in \mathbf{D}$ corresponds to the atomic concept D_i.

$$A_0(x) \rightarrow O_0(x) \tag{44}$$

$$O_0(x) \wedge R_1(x,y) \rightarrow O_0(y) \tag{45}$$

$$A_n(x) \wedge O_{i-1}(x) \wedge H(x,y) \rightarrow O_i(y) \text{ for each } 1 \leq i < n \tag{46}$$

$$A_n(x) \wedge O_i(x) \rightarrow D_{\alpha_i}(x) \text{ for each } 1 \leq i < n \tag{47}$$

$$A_n(x) \rightarrow D_1(x) \vee \ldots \vee D_m(x) \tag{48}$$

$$D_i(x) \wedge D_j(x) \rightarrow \bot \text{ for each } 1 \leq i < j \leq m \tag{49}$$

$$D_i(x) \wedge H(x,y) \wedge D_j(y) \rightarrow \bot \text{ for each } (D_i, D_j) \notin \mathbf{H} \tag{50}$$

$$D_i(x) \wedge V(x,y) \wedge D_j(y) \rightarrow \bot \text{ for each } (D_i, D_j) \notin \mathbf{V} \tag{51}$$

Let I be a model of $\mathcal{K}_{\mathbf{S},\mathbf{I}}$. By rules (44)–(45), the apex of the pyramid and each vertex that is reachable from the apex by an R_1-chain is labeled with O_0. Rule (46) ensure that the "first" n vertices in the n-th level of the pyramid are labeled with $O_0, \ldots, O_{n-1}$. Rules (47) label these vertices with the appropriate tiles from the initial condition. Finally, rules (48)–(51) ensure that each element at the n-th level of the grid is labeled with exactly one tile according to the compatibility conditions of $\mathbf{S}$. Therefore, $\mathcal{K}_{\mathbf{S},\mathbf{I}}$ is satisfiable if and only if a bounded $\mathbf{S}$-tiling for $\mathbf{I}$ exists, which proves our claim. □

Note that rule (48) contains a disjunction. Without disjunctions in the rules and description graphs (i.e., if for each graph G in the GBox of $\mathcal{K}$ we have that $\mid V_A \mid \leq 1$ for each main concept A of G), reasoning with $\mathcal{G}$ and $\mathcal{P}$ becomes deterministic and the complexity drops to ExpTime: the description graphs in $\mathcal{G}$ then encode a structure that can be computed deterministically in exponential time by unfolding $\mathcal{G}$, and the rules in $\mathcal{P}$ can be applied to this structure in exponential time as well. Axioms of the form (48), however, are available even in the basic description logic

$\mathcal{ALC}$, so the proof of Lemma 7.1 shows that reasoning with graph-extended KBs is NExpTime-hard even for basic DLs.

7.2 Upper Bounds

The hypertableau procedure from Section 6 is not worst-case optimal even without description graphs and rules, and with $\mathcal{T}$ in $\mathcal{ALC}$ [32, Section 5.3]. This is because an ABox $\mathcal{A}'$ labeling a derivation node for a set of HT-rules $\mathcal{R} = \Xi_{\mathcal{T}}(\mathcal{T})$ and an ABox $\Xi_{\mathcal{A}}(\mathcal{T}) \cup \mathcal{A}$ can at any given point in time contain at most exponentially many nonblocked and directly blocked tree individuals; however, $\mathcal{A}'$ can contain a doubly exponential number of indirectly blocked individuals. The complexity of the hypertableau procedure can be reduced to NExpTime if we ensure that the label of each individual s is fully determined in $\mathcal{A}'$ before applying the $\geq$-rule to an assertion containing s: the rule application strategy then ensures that s cannot subsequently become blocked, so $\mathcal{A}'$ never contains indirectly blocked individuals. A similar approach was used in [14] to obtain a tableau algorithm for $\mathcal{ALC}$ running in NExpTime. For $\mathcal{SHOQ}^+$ and $\mathcal{SHIQ}^+$, such an algorithm is not worst-case optimal, since these DLs are ExpTime-complete [3]. Description graphs increase the complexity at least to NExpTime, so the "excess" complexity of the modified hypertableau algorithm is not relevant: we next present two modified hypertableau algorithms that decide the satisfiability of a graph-extended acyclic knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ in NExpTime if $\mathcal{K}$ is acyclic and weakly separated with $\mathcal{T}$ in $\mathcal{SHOQ}^+$, or if $\mathcal{K}$ is acyclic and strongly separated with $\mathcal{T}$ in $\mathcal{SHIQ}^+$. Since they use extensive guessing to realize the idea outlined above, these algorithms are unlikely to be practicable.

Our modified algorithms can be applied to any $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ where $\mathcal{R}$ is normalized according to the following definition. This assumption is without loss of generality, since each set of HT-rules can be normalized by replacing concepts of the form $\geq n\, R.B$ in the rules with fresh atomic concepts.

Definition 18 (Normalized Rules)**.** *A set of HT-rules $\mathcal{R}$ is* normalized *if all at-least restriction concepts occur in $\mathcal{R}$ only in rules of the form* (52)*.*

$$A(x) \rightarrow \geq n\, R.B(x) \tag{52}$$

We are now ready to present the algorithm for the case when $\mathcal{T}$ is in $\mathcal{SHOQ}^+$.

Theorem 6. *Checking the satisfiability of a weakly separated acyclic graph-extended knowledge base $\mathcal{K}$ whose TBox is in $\mathcal{SHOQ}^+$ is* NExpTime*-complete, provided that the numbers in $\mathcal{K}$ are coded in unary.*

Proof. Hardness is shown in Lemma 3. By the properties of preprocessing [32], $\mathcal{K}$ can be transformed to an equisatisfiable triple $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ where $\mathcal{R}$ is normalized and $\mathcal{A}$ is an input ABox. We next show that the satisfiability of $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ can be decided by the following variant of the calculus from Section 6.

(1) The $\geq$-rule is modified such that, after it is applied to an assertion $\geq n\, R.B(s)$ in an ABox $\mathcal{A}'$ and it introduces fresh tree successors $t_1, \ldots, t_n$ of s, it also nondeterministically derives the following assertions for each $1 \leq i \leq n$:
   (a) $t_i \approx a$ or $t_i \not\approx a$ for each root individual a in $\mathcal{A}'$,
   (b) $t_i \approx u$ or $t_i \not\approx u$ for each individual u in $\mathcal{A}'$ from the cluster of s,
   (c) $t_i \approx u$ or $t_i \not\approx u$ for each tree successor u of s in $\mathcal{A}'$,
   (d) $t_i \approx t_j$ or $t_i \not\approx t_j$ for each $i < j \leq n$, and
   (e) $A(t_i)$ or $\neg A(t_i)$ for each atomic concept A occurring in $(\mathcal{R}, \mathcal{G}, \mathcal{A})$.

(2) The $G_\exists$-rule is modified such that, after it is applied to an assertion $A(s)$ in an ABox $\mathcal{A}'$ and it introduces fresh graph successors $t_1, \ldots, t_n$ of s, it also nondeterministically derives the following assertions for each $1 \leq i \leq n$:
   (a) $t_i \approx a$ or $t_i \not\approx a$ for each root individual a in $\mathcal{A}'$,
   (b) $t_i \approx u$ or $t_i \not\approx u$ for each individual u in $\mathcal{A}'$ from the cluster of t_i, and
   (c) $A(t_i)$ or $\neg A(t_i)$ for each atomic concept A occurring in $(\mathcal{R}, \mathcal{G}, \mathcal{A})$.

(3) When the $\approx$-rule is applied to an assertion $t_i \approx u$ derived in the previous two cases, it merges t_i into u.

(4) The rule precedence satisfies the following restrictions in addition to the ones given in Definition 15:
   (a) the $\bot$-rule is applied with the highest priority,
   (b) the $\approx$-rule is applied with the second-highest priority,
   (c) the *Hyp*-rule is applied to an HT-rule of the form (52) with the third-highest priority, and
   (d) the $\geq$-rule is applied only if the $G_\exists$-rule is not applicable.

(5) The strict ordering $<$ used in the definition of anywhere blocking follows the creation order—that is, if an individual s is added to an ABox before an individual t, then $s < t$.

The modified $\geq$- and $G_\exists$-rules are obviously sound, so the proof of Claim 1 of Theorem 1 applies with minor changes. Furthermore, all assertions introduced by the $\geq$- and $G_\exists$-rules are of the form as specified in Lemma 1, and the new rule precedence is stronger than the one in Definition 15; hence, the proofs of Claims 2 and 3 of Theorem 1 apply without any change.

Let $\mathcal{A}_0$ be an ABox labeling a node in a derivation for $(\mathcal{R}, \mathcal{G}, \mathcal{A})$; let $\mathcal{A}_1$ be an ABox obtained from $\mathcal{A}_0$ by an application of the $\geq$- or $G_\exists$-rule that introduces fresh individuals $t_1, \ldots t_n$; and let $\mathcal{A}_2$ be a clash-free ABox obtained from $\mathcal{A}_1$ by exhaustive applications of the rules mentioned in Items (4a)–(4c).

We now show the following property ($\bigstar$): if $\mathcal{A}_2$ contains t_i, $\mathcal{A}_3$ is a clash-free ABox obtained from $\mathcal{A}_2$ by applying one or more derivation rules, and the $\bot$-rule is not applicable to $\mathcal{A}_3$, then $\mathcal{A}_3$ contains t_i as well and $\mathcal{L}_{\mathcal{A}_2}(t_i) = \mathcal{L}_{\mathcal{A}_3}(t_i)$. This follows from the following facts:

- By Items (1a)–(1d), (2a)–(2b), and (4a)–(4b), ABox $\mathcal{A}_2$ contains an inequality $t_i \not\approx u$ for each individual u from $\mathcal{A}_2$ that t_i could potentially be merged into. Hence, if $\mathcal{A}_3$ is derived by merging t_i into some individual from $\mathcal{A}_2$, then $\mathcal{A}_3$ contains $\bot$.
- By Item (3), if $\mathcal{A}_3$ is derived by an application of the $\approx$-rule to $t_i \approx u$, then u is merged into t_i, so $\mathcal{A}_3$ contains t_i.
- By Items (1e) and (2c), ABox $\mathcal{A}_2$ contains either $A(t_i)$ or $\neg A(t_i)$ for each atomic concept A. Hence, $\mathcal{A}_3$ is derived by adding an assertion of the form $B(t_i)$, then $\mathcal{A}_3$ contains $\bot$.
- By Item (4c) and the fact that $\mathcal{R}$ is normalized, $\mathcal{A}_2$ contains all assertions of the form $\geq n\, R.B(t_i)$ implied by the HT-rules of the form (52). Since no derivation rule can introduce a concept $A(t_i)$ without introducing a clash, no HT-rule of the form (52) can be used to derive a new assertion of the form $\geq n\, R.B(t_i)$.
- By Items (1e), (2c), and (4d), the $G_\exists$-rule is applied exhaustively to individuals in the cluster of t_i before the $\geq$-rule can introduce a tree descendant of $\lfloor t_i \rfloor$. Therefore, if a subsequent application of the $\geq$-rule introduces an individual v, either $t_i \approx v$ or $t_i \not\approx v$ will be introduced by Item (1b), which allows for an inductive application of this argument.

We also show the following property ($\blacklozenge$): no individual in $\mathcal{A}_3$ is indirectly blocked. This is because ($\bigstar$) implies that the blocking status of t_i is the same in $\mathcal{A}_2$ and $\mathcal{A}_3$, which means that no descendant of t_i can become indirectly blocked by t_i.

We also show the following property ($\lozenge$): pruning never removes individuals from an ABox. This is a simple consequence of the fact that, by ($\bigstar$), Items (3) and (4a)–(4d), individual t_i is either merged into an individual from $\mathcal{A}_1$ in the derivation of $\mathcal{A}_2$, or it cannot participate in merging inference used to derive $\mathcal{A}_3$.

Let c be the number of atomic concepts in $(\mathcal{R}, \mathcal{G}, \mathcal{A})$; by Item (5) and the definition of single blocking, $\mathcal{A}_0$ can contain at most 2^c nonblocked tree individuals. As shown in the proof of Claim 2 of Theorem 1, $\mathcal{A}_0$ can contain the tree cluster of each tree individual can contain at most ϑ graph individuals. Furthermore, ($\blacklozenge$) implies that $\mathcal{A}_0$ can contain at most exponentially many blocked individuals.

Consider now a named individual a in $\mathcal{A}_0$. Since individuals are reused in graph cleanup, merging a graph individual u into a can introduce at most ϑ individuals for each label $\omega(u)$ of the form (35). There are at most ϑ such labels, so graph cleanup can introduce at most ϑ^2 individuals for a. In the same way as in the proof of Item (3) of Claim 2 of Theorem 1, the $G_\exists$-rule can introduce at most ϑ individuals for each of the ϑ^2 individuals. Thus, $\mathcal{A}_0$ contains at most $\vartheta_r = i \cdot \vartheta^3$ root graph individuals, where i is the number of named individuals in $(\mathcal{R}, \mathcal{G}, \mathcal{A})$.

Thus, the total number of individuals i_{tot} in each clash-free ABox is at most exponential in $(\mathcal{R}, \mathcal{G}, \mathcal{A})$. Furthermore, each derivation rule is applied to a polynomial number of individuals. Finally, by ($\lozenge$), individuals are never removed from an ABox by pruning. Thus, a derivation path for $(\mathcal{R}, \mathcal{G}, \mathcal{A})$ can be constructed in nondeterministic exponential time, which implies our claim. □

We now prove an analogous claim for the case when $\mathcal{T}$ is in $\mathcal{SHIQ}^+$.

Theorem 7. *Checking the satisfiability of a strongly separated acyclic graph-extended knowledge base $\mathcal{K}$ whose TBox is in $\mathcal{SHIQ}^+$ is* NExpTime*-complete, provided that the numbers in $\mathcal{K}$ are coded in unary.*

Proof. Hardness is shown in Lemma 3, and the membership proof is analogous to Theorem 6. The hypertableau algorithm from Section 6 is modified as follows:

(1) The $\geq$-rule is modified such that, after it is applied to an assertion $\geq n\, R.B(s)$ in an ABox $\mathcal{A}'$ and it introduces fresh tree successors $t_1, \ldots, t_n$ of s, it also nondeterministically derives the following assertions for each $1 \leq i \leq n$:
   (a) $t_i \approx u$ or $t_i \not\approx u$ if $\mathcal{A}'$ contains a tree predecessor u of s,
   (b) $t_i \approx s$ or $t_i \not\approx s$,
   (c) $t_i \approx u$ or $t_i \not\approx u$ for each tree successor u of s in $\mathcal{A}'$,
   (d) $t_i \approx t_j$ or $t_i \not\approx t_j$ for each $i < j \leq n$,
   (e) $A(t_i)$ or $\neg A(t_i)$ for each atomic concept A occurring in $(\mathcal{R}, \mathcal{G}, \mathcal{A})$, and
   (f) $R(s, t_i)$ or $\neg R(s, t_i)$, as well as $R(t_i, s)$ or $\neg R(t_i, s)$, for each atomic $\mathcal{T}$-role R occurring in $(\mathcal{R}, \mathcal{G}, \mathcal{A})$.

(2) The $G_\exists$-rule is modified such that, after it is applied to an assertion $A(s)$ in an ABox $\mathcal{A}'$ and it introduces fresh graph successors $t_1, \ldots, t_n$ of s, it also nondeterministically derives the following assertions for each $1 \leq i \leq n$:
   (a) $t_i \approx u$ or $t_i \not\approx u$ for each individual u in $\mathcal{A}'$ from the cluster of t_i, and
   (b) $A(t_i)$ or $\neg A(t_i)$ for each atomic concept A occurring in $(\mathcal{R}, \mathcal{G}, \mathcal{A})$.

(3) The $\bot$-rule is amended to derive $\bot$ if $\mathcal{A}'$ contains both $R(s, t)$ and $\neg R(s, t)$.

(4) Items (2)–(5) from the proof of Theorem 7 are used without change.

Although negative role assertions of the form $\neg R(u, v)$ do not satisfy Lemma 2, they do not participate in the model construction from the proof of Theorem 3; therefore, Theorem 4 holds without any change. Furthermore, Item (1e) of the modified calculus ensures that, for each individual s and each tree successor t of s, labels $\mathcal{L}_\mathcal{A}(s, t)$ and $\mathcal{L}_\mathcal{A}(t, s)$ are fully determined after the application of the $\geq$-rule; thus, an individual t_j becomes pairwise-blocked immediately after it has been introduced, or it never becomes blocked. The rest of the argument is analogous to the proof of Theorem 6. □

8 Implementation

We have implemented our reasoning algorithm in the hypertableau-based reasoner HermiT [32]. Evaluating the adequacy of our approach is rather difficult due to lack of 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 $\mathcal{T}_1$ into a graph-extended knowledge base $\mathcal{K}$. The knowledge base $\mathcal{K}$, even if only a rough approximation, can be used as a starting point for a more comprehensive remodeling of $\mathcal{T}_1$ into a proper graph-extended KB. Our experience with GALEN and the discussions we had with the authors of GALEN led us to conclude that the transformed KB $\mathcal{K}$ represents the anatomical structures in the human body in a way that is closer to the modelers' intention than the original DL axioms. [10]

8.1 The Transformation Algorithm

Our algorithm transforms a TBox $\mathcal{T}_1$ into a graph-extended $\mathcal{K} = (\mathcal{T}, \mathcal{G}, \mathcal{P}, \mathcal{A})$ such that $\mathcal{K}$ is strongly-separated and $\mathcal{G}$ contains only one description graph. It would clearly be more useful if we could automatically transform $\mathcal{T}_1$ into several smaller description graphs; however, it was unclear what kinds of heuristics to use in order to determine the boundaries between different description graphs.

Our transformation is based on two assumptions. The first assumption is that only some concepts and roles from $\mathcal{T}_1$ are relevant to G. For example, *Hand* is relevant to the graph of the human body, but *Fracture* is not; similarly, the *hasPart* role belongs to the graph, while the *hasAge* role does not. The second assumption is that each relevant concept should be represented by one vertex in G, and edges in G can be decoded from axioms of the form $A \sqsubseteq \exists R.B$. In other words, we conjecture that, by writing axioms such as (53)–(55), modelers actually wanted to say "the index finger has a middle phalanx and a proximal phalanx as parts, and these two phalanges are attached to each other."

$$\mathit{Index\_finger} \sqsubseteq \exists \mathit{part}.\mathit{Middle\_phalanx\_oif} \tag{53}$$

$$\mathit{Distal\_phalanx\_oif} \sqsubseteq \exists \mathit{attached\_to}.\mathit{Middle\_phalanx\_oif} \tag{54}$$

$$\mathit{Proximal\_phalanx\_oif} \sqsubseteq \exists \mathit{part}^{-}.\mathit{Index\_finger} \tag{55}$$

Our algorithm is given 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 $\mathcal{G}$-roles, and all other roles are $\mathcal{T}$-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 in N_{R_g}, then α is copied to $\mathcal{T}$.
- 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 $\mathcal{G}$-role and a $\mathcal{T}$-role simultaneously, or it is of the form $A \sqsubseteq \exists R.B$ but some of A, B, or R are not

[10] Thanks to Alan Rector and Sebastian Brandt.

Table 6
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
Translated GCIs:	6642	3151
Into the description graph:	680	2966
Into rules over the graph:	155	1
Into the DL TBox	5807	184
Vertices in the description graph:	325	342
Edges in the description graph:	667	1076

relevant for the graph. Such an axiom either invalidates the syntactic restrictions of our formalism or it does not have a natural interpretation. Human intervention would be needed to interpret such axioms in a "reasonable" way; therefore, such axioms are discarded by our algorithm.

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.

8.2 Classification Results

To evaluate our approach, we have classified the original version of GALEN and a fragment of FMA. Next, we have transformed them into graph-extended KBs, and classified the resulting KBs using the reasoning algorithm presented in Section 6. We now discuss the obtained results. Table 6 summarizes information about the original and the transformed ontologies.

We performed the experiments using a standard laptop with 1 GB of RAM. Classification of the original version of GALEN and the fragment of FMA took 129 s and 57 s, respectively; furthermore, classification of the transformed ontologies took 781 s and 6 s, respectively. The increase in classification time for GALEN is partly due to the prototypical nature of our implementation. In the case of FMA, classification times are substantially lower because most of the original ontology is translated into the graph, so the generated models are much smaller. Our results show that, even with a very prototypical implementation, complex ontologies can be processed, which we take as indication that our approach is practically feasible.

Our transformation leads to a change in the semantics of the ontology: some axioms are lost in the process, and the semantics of many axioms is modified. Many parts of the resulting description graph, however, correspond with an intuitive

description of the human body. For example, the (union of) the graphs shown in Figures 2b–2e has been extracted from the transformed ontology.

Although some information is lost in the translation, the resulting description graphs can be seen as being "more precise" than the original axioms, so one can expect to obtain new entailments. For example, we discovered in GALEN a concept that is satisfiable in the original ontology, but is unsatisfiable in the transformed ontology; this revealed a modeling error in the original ontology. The problem occurs in the representation of the patella—a bone in a knee 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:

$$LateralPatellaRetinaculum \equiv \exists hasOtherEndAt.Patella \sqcap (\ldots) \tag{56}$$

$$MedialPatellaRetinaculum \equiv \exists hasOtherEndAt.Patella \sqcap (\ldots) \tag{57}$$

$$hasOtherEndAt \equiv isAtOtherEndOf^{-} \tag{58}$$

$$\top \sqsubseteq \leq 1\, isAtOtherEndOf \tag{59}$$

In a human body, each patella is connected to a lateral and a medial retinaculum. In GALEN, however, *isAtOtherEndOf* is functional, so the two retinacula connected to a patella must always be one and the same object. This leads us to believe that *isAtOtherEndOf* probably should not have been declared functional. GALEN, however, is underconstrained: it does not require the lateral retinaculum and the medium retinaculum of a knee to be connected to the same patella, and it does not state that the lateral retinaculum and the medial retinaculum are different from each other. Consequently, the concept *Patella* is consistent in GALEN, and this modeling error was not detected. 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 of description graphs (see Definition 10) and makes *Patella* unsatisfiable.

9 Conclusion

We have presented an expressive formalism that extends DLs with description graphs and rules, allowing for more precise modeling of arbitrarily connected structures. Our formalism is applicable not only to anatomy, but to all domains in which the number of arbitrarily interconnected objects has a natural bound.

The main open theoretical challenges are to determine the decidability and/or complexity of reasoning with graph-extended knowledge bases under different assumptions on the expressivity of the DL TBox $\mathcal{T}$ and the set of rules $\mathcal{P}$. All our undecidability results from Section 5 require $\mathcal{T}$ to contain number restrictions. We conjecture that if $\mathcal{T}$ is not allowed to contain number restrictions, $\mathcal{P} = \emptyset$, and $\mathcal{G}$ does not contain graph alignments, then reasoning becomes decidable even if $\mathcal{G}$ is

not acyclic. This is because such $\mathcal{T}$ and the properties from Table 4 apart from the key and the disjointness properties can be transformed into an equivalent formula of the guarded fragment of first-order logic which is known to be decidable [2]; furthermore, the key and the disjointness properties seem "innocuous" because they merely prevent an axiomatization of infinite chains of instances of one description graph. Another important research direction is to see whether decidability can be achieved by placing different restrictions on the set of rules $\mathcal{P}$. For example, we conjecture that, even without any role separation requirement, our formalism can be extended with ELP rules [24] without losing decidability. Finally, the complexity of reasoning with a strongly separated and acyclic graph extended knowledge base whose TBox is in $\mathcal{SHOIQ}^+$ is open.

The main practical challenge is to validate the applicability of our formalism in these and other applications. To this end, we will extend the ontology editor Protégé 4 to support description graphs and apply our formalism in the identified practical scenarios.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison Wesley, 1995.

[2] H. Andréka, J. van Benthem, and I. Németi. Modal Languages and Bounded Fragments of Predicate Logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.

[3] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2nd edition, August 2007.

[4] F. Baader and U. Sattler. An Overview of Tableau Algorithms for Description Logics. *Studia Logica*, 69:5–40, 2001.

[5] Franz Baader, Carsten Lutz, Holger Sturm, and Frank Wolter. Fusions of Description Logics and Abstract Description Systems. *Journal of Artificial Intelligence Research*, 16:1–58, 2002.

[6] P. Baumgartner, U. Furbach, and I. Niemelä. Hyper Tableaux. In *Proc. of the European Workshop on Logics in Artificial Intelligence (JELIA '96)*, number 1126 in LNAI, pages 1–17, Évora, Portugal, September 30–October 3 1996. Springer.

[7] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Springer, 1996.

[8] A. Borgida. On the Relative Expressiveness of Description Logics and Predicate Logics. *Artificial Intelligence*, 82(1–2):353–367, 1996.

[9] A. Calì, G. Gottlob, and M. Kifer. Taming the Infinite Chase: Query Answering under Expressive Relational Constraints. In Gerhard Brewka and Jérôme Lang, editors, *Proc. of the 11th Int. Joint Conf. on Principles of Knowledge Representation and Reasoning (KR 2008)*, pages 70–80, Sydney, NSW, Australia, August 16–19 2008. AAAI Press.

[10] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider, and Ulrike Sattler. OWL 2: The next step for OWL. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):309–322, 2008.

[11] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Computing Surveys*, 33(3):374–425, 2001.

[12] J. de Bruijn, A. Polleres, R. Lara, and D. Fensel. OWL DL vs. OWL Flight: Conceptual Modeling and Reasoning on the Semantic Web. In *Proc. of the 14th Int. World Wide Web Conference (WWW2005)*, pages 623–632, Chiba, Japan, May 10–14 2005. ACM.

[13] S. Demri and H. de Nivelle. Deciding Regular Grammar Logics with Converse Through First-Order Logic. *Journal of Logic, Language and Information*, 14(3):289–329, 2005.

[14] 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.

[15] T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining Answer Set Programming with Description Logics for the Semantic Web. In D. Dubois, C. A. Welty, and M.-A. Williams, editors, *Proc. of the 9th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 141–151, Whistler, Canada, June 2–5, 2004 2004. AAAI Press.

[16] A. Goderis, U. Sattler, and C. A. Goble. Applying Description Logics for Workflow Reuse and Repurposing. In Ian Horrocks, Ulrike Sattler, and Frank Wolter, editors, *Proc. of the 2005 Int. Workshop on Description Logics (DL 2005)*, volume 147 of *CEUR Workshop Proceedings*, Edinburgh, UK, July 26–28 2005.

[17] H. Graves. Representing Product Designs Using a Description Graph Extension to OWL 2. In Alan Ruttenberg, Ulrile Sattler, and Cathy Dolbear, editors, *Proc. of the 5th Int. Workshop on OWL: Experiences and Directions (OWLED 2008 EU)*, Karlsruhe, Germany, October 26–27 2008.

[18] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation (2nd Edition)*. Addison Wesley, 2nd edition, November 24 2000.

[19] I. Horrocks and H. Graves. Application of OWL 1.1 to Systems Engineering. In K. Clark and P. F. Patel-Schneider, editors, *Proc. of the OWL: Expreiences and Directions Workshop (OWLED 2008 DC)*, Washington, DC, USA, April 1–2 2008.

[20] I. Horrocks and P. F. Patel-Schneider. A Proposal for an OWL Rules Language. In *Proc. of the 13th Int. World Wide Web Conference (WWW 2004)*, pages 723–731, New York, NY, USA, May 17–22 2004. ACM Press.

[21] I. Horrocks and U. Sattler. Decidability of $\mathcal{SHIQ}$ with Complex Role Inclusion Axioms. *Artificial Intelligence*, 160(1–2):79–104, December 2004.

[22] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42(4):741–843, 1995.

[23] M. Konyk, A. De Leon Battista, and Dumontier M. Chemical Knowledge for the Semantic Web. In *Proc. of the 5th Int. Workshop on Data Integration in the Life Sciences (DILS 2008)*, volume 5109 of *LNCS*, pages 169–176. Springer, 2008.

[24] M. Krötzsch, S. Rudolph, and P. Hitzler. ELP: Tractable Rules for OWL 2. In Amit P. Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, and Krishnaprasad Thirunarayan, editors, *Proc. of the 7th Int. Semantic Web Conference (ISWC 2008)*, volume 5318 of *LNCS*, pages 649–664, Karlsruhe, Germany, October 26–30 2008. Springer.

[25] O. Kutz, I. Horrocks, and U. Sattler. The Even More Irresistible $\mathcal{SROIQ}$. In P. Doherty, J. Mylopoulos, and C. A. Welty, editors, *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 68–78, Lake District, UK, June 2–5 2006. AAAI Press.

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

[27] B. Motik. *Reasoning in Description Logics using Resolution and Deductive Databases*. PhD thesis, Univesität Karlsruhe, Germany, 2006.

[28] B. Motik, B. Cuenca Grau, I. Horrocks, and U. Sattler. Representing Structured Objects using Description Graphs. In Gerhard Brewka and Jérôme Lang, editors, *Proc. of the 11th Int. Joint Conf. on Principles of Knowledge Representation and Reasoning (KR 2008)*, pages 296–306, Sydney, NSW, Australia, August 16–19 2008. AAAI Press.

[29] B. Motik, B. Cuenca Grau, and U. Sattler. Structured Objects in OWL: Representation and Reasoning. In *Proc. of the 17th Int. World Wide Web Conference (WWW 2008)*, pages 555–564, Beijing, China, April 21–25 2008. ACM Press.

[30] B. Motik and R. Rosati. A Faithful Integration of Description Logics with Logic Programming. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI 2007)*, pages 477–482, Hyderabad, India, January 6–12 2007. Morgan Kaufmann Publishers.

[31] B. Motik, R. Shearer, and I. Horrocks. Optimized Reasoning in Description Logics using Hypertableaux. In F. Pfenning, editor, *Proc. of the 21st Conference on Automated Deduction (CADE-21)*, volume 4603 of *LNAI*, pages 67–83, Bremen, Germany, July 17–20 2007. Springer.

[32] B. Motik, R. Shearer, and I. Horrocks. Hypertableau Reasoning for Description Logics. Technical report, University of Oxford, 2008. Submitted to an international journal.

[33] Boris Motik, Ulrike Sattler, and Rudi Studer. Query Answering for OWL-DL with Rules. *Journal of Web Semantics*, 3(1):41–60, 2005.

[34] J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis. Telos: Representing Knowledge about Information Systems. *ACM Transactions of Information Systems*, 8(4):325–362, 1990.

[35] P. F. Patel-Schneider, P. Hayes, and I. Horrocks. OWL Web Ontology Language: Semantics and Abstract Syntax, W3C Recommendation, February 10 2004. `http://www.w3.org/TR/owl-semantics/`.

[36] D. A. Plaisted and S. Greenbaum. A Structure-Preserving Clause Form Translation. *Journal of Symbolic Logic and Computation*, 2(3):293–304, 1986.

[37] R. Rosati. $\mathcal{DL}$+*log*: A Tight Integration of Description Logics and Disjunctive Datalog. In P. Doherty, J. Mylopoulos, and C. A. Welty, editors, *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 68–78, Lake District, UK, June 2–5 2006. AAAI Press.

[38] R. A. Schmidt and U. Hustadt. A Principle for Incorporating Axioms into the First-Order Translation of Modal Formulae. In F. Baader, editor, *Proc. of the 19th Int. Conf. on Automated Deduction (CADE-19)*, volume 2741 of *LNAI*, pages 412–426, Miami Beach, FL, USA, July 28–August 2 2003. Springer.

[39] S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, Germany, 2001.

[40] M. Y. Vardi. Why Is Modal Logic So Robustly Decidable? In N. Immerman and P. Kolaitis, editors, *Proc. of a DIMACS Workshop on Descriptive Complexity and Finite Models*, volume 31 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 149–184, Princeton University, USA, January 14–17 1996. American Mathematical Society.