Conservative extensions in Expressive Description Logics

Carsten Lutz, Dirk Walther, Frank Wolter

Abstract

The notion of a conservative extension plays a central role in ontology design and integration: it can be used to formalize ontology refinements, safe mergings of two ontologies, and independent modules inside an ontology. Regarding reasoning support, the most basic task is to decide whether one ontology is a conservative extension of another. It has recently been proved that this problem is decidable and 2ExpTime-complete if ontologies are formulated in the basic description logic (DL) $\mathcal{ALC}$. We consider more expressive DLs and begin to map out the boundary between decidable and undecidable by proving that conservative extensions are 2ExpTime-complete in $\mathcal{ALCQI}$, but undecidable in $\mathcal{ALCQIO}$. We also show that if conservative extensions are defined model-theoretically rather than in terms of the consequence relation, they are undecidable already in $\mathcal{ALC}$.

1 Introduction

The design and integration of ontologies formulated in modern ontology languages such as OWL is a serious challenge. Experience shows that principled methodologies as well as automated reasoning support are required to ensure that the resulting ontologies are well-structured [6]. In the recent papers [4, 5, 3, 1], conservative extensions have been identified as a crucial notion for formalizing central tasks in ontology design and integration. Consequently, conservative extensions can play a key role in design and integration methodologies [3], provided that reasoning tools are available which support the ontology designer in the task of using this notion.

Formally, an ontology $T_1 \cup T_2$ is a conservative extension of an ontology T_1 w.r.t. a signature Γ iff every consequence of $T_1 \cup T_2$ formulated in Γ is already a consequence of T_1. For example, assume that ontologies are formalized in a description logic (DL) such as OWL-DL and its fragments [2]. Then a signature is a set of concept and role names, an ontology is a TBox, and a consequence of a TBox is a subsumption relationship between two concepts which follows from the TBox. Intuitively, $T_1 \cup T_2$ is a conservative extension of T_1 w.r.t. Γ if adding T_2 to T_1 does not change the ontology T_1 as far as concepts built only from concept and role names in Γ are concerned. We give three examples of ontology related tasks that can be understood in terms of conservative extensions.

– *Ontology refinement*. During ontology design, a frequent task is to add more details to a part of the ontology that has not yet been sufficiently described. Intuitively, such a refinement should have no impact on other, unrelated parts of the ontology. This requirement can be formalized by demanding that the refined ontology is a conservative extension of the original one w.r.t. the concept and role names that do not belong to the refined part [4, 1].

– *Ontology merging*. The most straightforward way to integrate two ontologies is to simply take their union. Such a merging should not compromise the original ontologies. One possible formalization of this requirement is to demand that the united ontology is a conservative extension of the component ontologies w.r.t. the set of all concept and role names used in the respective components. Weaker formalizations are obtained by excluding from the signature concept and role names for which an interaction between the component ontologies is expected (and intended).

– *Defining Modules*. A module inside an ontology T that describes an independent part of the application domain can be defined as a subset T' of T such that T is a conservative extension of T' w.r.t. the concept names and role names that belong to T' [5].

The most basic reasoning task regarding conservative extensions is as follows: given ontologies T_1 and T_2 and a signature Γ, decide whether $T_1 \cup T_2$ is a conservative extension of T_1 w.r.t. Γ. In the following, we refer to this task as *deciding conservative extensions*. [4] carries out a first investigation of this decision problem for DLs, showing in particular that this problem is decidable and 2-ExpTime-complete for the basic description logic $\mathcal{ALC}$. The aim of the current paper is to *investigate conservative extensions in more expressive DLs such as the ones underlying the ontology language OWL-DL and to map out the boundary between decidable and undecidable*. Our main results are as follows: (i) in $\mathcal{ALCQI}$, the extension of $\mathcal{ALC}$ with inverse roles and qualifying number restrictions, deciding conservative extensions is 2-ExpTime complete and thus not more difficult than in $\mathcal{ALC}$; and (ii) if we further extend $\mathcal{ALCQI}$ with nominals, conservative extensions in the resulting DL $\mathcal{ALCQIO}$ are undecidable. This shows that conservative extensions in OWL-DL, of which $\mathcal{ALCQIO}$ is a fragment, is also undecidable. It also identifies $\mathcal{ALCQI}$ as a significant fragment of OWL-DL in which conservative extensions are still decidable.

In mathematical logic, there exist (at least) two versions of conservative extensions. One is based on the consequence relation as sketched above. An alternative, stronger version is defined in a model-theoretic way. We also consider deciding the latter kind of conservative extensions and show that, already in $\mathcal{ALC}$, this problem is highly undecidable.

Name	Syntax	Semantics
inverse role	r^-	$(r^\mathcal{I})^{-1}$
nominal	$\{a\}$	$\{a^\mathcal{I}\}$
negation	$\neg C$	$\Delta^\mathcal{I} \setminus C^\mathcal{I}$
conjunction	$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
at-most number restriction	$(\leqslant n\ r\ C)$	$\{d \mid \#\{e \mid (d,e) \in r^\mathcal{I} \wedge e \in C^\mathcal{I}\} \leq n\}$

Figure 1: Syntax and semantics of $\mathcal{ALCQIO}$.

2 Preliminaries

In DLs, *concepts* are inductively defined with the help of a set of *constructors*, starting with a set $\mathsf{N_C}$ of *concept names*, a set $\mathsf{N_R}$ of *role names*, and (possibly) a set $\mathsf{N_I}$ of *individual names*. The constructors available in $\mathcal{ALCQIO}$ are shown in Figure 1. There, the inverse constructor is the only role constructor, whereas the remaining constructors are concept constructors. In Figure 1 and the remainder of this paper, we use $\#S$ to denote the cardinality of a set S, a and b to denote individual names, r to denote roles (i.e., role names and inverses thereof), A, B to denote concept names, and C, D to denote (possibly complex) concepts. For an inverse role $s = r^-$ we set $s^- = r$. As usual, we use $\top$ as abbreviation for an arbitrary (but fixed) propositional tautology, $\bot$ for $\neg\top$, $\sqcup$, $\rightarrow$, and $\leftrightarrow$ for the usual Boolean abbreviations, $(\geqslant n\ r\ C)$ (*at-least restriction*) for $\neg(\leqslant n-1\ r\ C)$ if $n > 0$ and for $\top$ if $n = 0$, $(= n\ r\ C)$ for $(\leqslant n\ r\ C) \sqcap (\geqslant n\ r\ C)$, $\exists r.C$ (*existential restriction*) for $(\geqslant 1\ r\ C)$, and $\forall r.C$ (*universal restriction*) for $(\leqslant 0\ r\ \neg C)$. We assume that the numbers inside number restrictions are coded in binary.

The DL that allows only for negation, conjunction, disjunction, and universal and existential restrictions is called $\mathcal{ALC}$. The availability of additional constructors is indicated by concatenation of a corresponding letter: $\mathcal{Q}$ stands for number restrictions; $\mathcal{I}$ stands for inverse roles, and $\mathcal{O}$ for nominals. This explains the name $\mathcal{ALCQIO}$ for our DL, and also allows us to refer to its sublanguages in a simple way.

The representation of ontologies in description logics is based on TBoxes, and we will from now on use these two terms interchangeably. Given a DL $\mathcal{L}$, an *$\mathcal{L}$-TBox* is a finite set of concept implications $C \sqsubseteq D$ with C and D $\mathcal{L}$-concepts.

The semantics of $\mathcal{ALCQIO}$-concepts is defined in terms of an *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$. The *domain* $\Delta^\mathcal{I}$ is a non-empty set of individuals and the *interpretation function* $\cdot^\mathcal{I}$ maps each concept name $A \in \mathsf{N_C}$ to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$, each role name $r \in \mathsf{N_R}$ to a binary relation $r^\mathcal{I}$ on $\Delta^\mathcal{I}$, and each individual name $a \in \mathsf{N_I}$ to an individual $a^\mathcal{I} \in \Delta^\mathcal{I}$. The extension of $\cdot^\mathcal{I}$ to inverse roles and arbitrary concepts is defined inductively as shown in the third column of Figure 1.

An interpretation $\mathcal{I}$ *satisfies* an implication $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ if it satisfies all implications in $\mathcal{T}$. A concept C is *satisfiable* relative to a TBox $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$. A concept C is *subsumed* by a concept D relative to a TBox $\mathcal{T}$ (written $\mathcal{T} \models C \sqsubseteq D$) if every model $\mathcal{I}$ of $\mathcal{T}$ satisfies the implication $C \sqsubseteq D$.

2.1 Conservative Extensions

Despite the fact that individual names are closer to constants than to predicates, we henceforth use the term *predicates* to refer to elements of $\mathsf{N_C} \cup \mathsf{N_R} \cup \mathsf{N_I}$. A *signature* is a finite set of predicates. The signature *sig*$(\mathcal{T})$ *of a TBox* $\mathcal{T}$ is the set of all predicates that occur in $\mathcal{T}$. Given a description logic $\mathcal{L}$ and a signature Γ, we use $\mathcal{L}(\Gamma)$ to denote the set of $\mathcal{L}$-concepts that use only predicates from Γ.

Definition 1 (Conservative Extension) *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be $\mathcal{L}$-TBoxes, and let $\Gamma \subseteq sig(\mathcal{T}_1)$ be a signature. Then $\mathcal{T}_1 \cup \mathcal{T}_2$ is* a Γ-conservative extension of *$\mathcal{T}_1$ if for all $C_1, C_2 \in \mathcal{L}(\Gamma)$, we have $\mathcal{T}_1 \models C_1 \sqsubseteq C_2$ iff $\mathcal{T}_1 \cup \mathcal{T}_2 \models C_1 \sqsubseteq C_2$.*

Deciding conservative extensions in $\mathcal{L}$ *means to decide, given two $\mathcal{L}$-TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ and a signature $\Gamma \subseteq sig(\mathcal{T}_1)$, whether $\mathcal{T}_1 \cup \mathcal{T}_2$ is a Γ-conservative extension of $\mathcal{T}_1$.*

If $\Gamma = \mathsf{sig}(\mathcal{T}_1)$, we simply drop Γ and only talk about conservative extensions. It is not difficult to see that an alternative definition of conservative extensions is as follows: $\mathcal{T}_1 \cup \mathcal{T}_2$ is a Γ-conservative extension of $\mathcal{T}_1$ iff each concept $C \in \mathcal{L}(\Gamma)$ that is satisfiable relative to $\mathcal{T}_1$ is satisfiable relative to $\mathcal{T}_1 \cup \mathcal{T}_2$. Therefore, a concept $C \in \mathcal{L}(\Gamma)$ that is satisfiable relative to $\mathcal{T}_1$, but not relative to $\mathcal{T}_1 \cup \mathcal{T}_2$ witnesses that $\mathcal{T}_1 \cup \mathcal{T}_2$ is *not* a conservative extension of $\mathcal{T}_1$. We call such a concept a Γ-*witness concept*.

Let us give an example for conservative extensions in the description logic $\mathcal{ALCQI}$. Assume that $\mathcal{T}_1$ is a TBox formalizing knowledge about universities:

$$\begin{array}{rcl}
\mathsf{Lecture} & \sqsubseteq & \exists \mathsf{has_subject.Subject} \sqcap \exists \mathsf{given_by.Lecturer} \\
\mathsf{Intro_TCS} & \sqsubseteq & \mathsf{Lecture} \\
\\
\mathsf{Lecturer} & \sqsubseteq & \mathsf{Professor} \sqcup \mathsf{TeachingAssistant} \\
\mathsf{Lecturer} & \sqsubseteq & \exists \mathsf{employed.University} \\
\mathsf{University} & \sqsubseteq & \forall \mathsf{employed}^{-}.(\mathsf{Academic} \sqcup \mathsf{Admin})
\end{array}$$

The upper part of $\mathcal{T}_1$ describes university lectures, saying e.g. that every introductory lecture on theoretical computer science (TCS) is a lecture. The lower part of $\mathcal{T}_1$ describes universities and their employees. Suppose now that we want to refine the part of the ontology that is concerned with lectures.

We extend the signature by adding the concept names $\mathsf{AutomataTheory}$ and $\mathsf{ComplexityTheory}$ and state that these subjects are discussed in every introductory TCS lecture. We also say that automata theory and complexity theory are different things:

$$\begin{array}{rcl} \mathsf{Intro\_TCS} & \sqsubseteq & \exists \mathsf{has\_subject}.\mathsf{AutomataTheory} \\ \mathsf{Intro\_TCS} & \sqsubseteq & \exists \mathsf{has\_subject}.\mathsf{ComplexityTheory} \\ \bot & \sqsupseteq & \mathsf{AutomataTheory} \sqcap \mathsf{ComplexityTheory} \end{array}$$

Intuitively, this addition should have an impact on the upper part of $\mathcal{T}_1$ since it adds information about lectures, but it should not affect the lower part which is not concerned with lectures. This intuition can be formally captured by conservative extensions: if we choose Γ to be the set of all predicates used in the lower part of $\mathcal{T}_1$, then $\mathcal{T}_1 \cup \mathcal{T}_2$ is a Γ-conservative extension of $\mathcal{T}_1$. Thus, the lower part of $\mathcal{T}_1$ is not affected by the addition of $\mathcal{T}_2$. If we choose Γ to be the predicates in the upper part of $\mathcal{T}_1$, then $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a Γ-conservative extension, a witness concept being

$$\mathsf{IntroTCS} \sqcap (\leqslant 1 \ \mathsf{has\_subject} \ \top).$$

By considering these two cases of conservative extensions, the ontology designer can thus verify that his modification changes the TBox (only) in the intended way.

This example also shows that conservative extensions depend on the descripion logic $\mathcal{L}$: the TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ are actually formulated in $\mathcal{ALCI}$ and we have seen that if Γ is the set of predicates in the upper part of $\mathcal{T}_1$, then $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a Γ-conservative extension of $\mathcal{T}_1$. However, this only holds since we assumed $\mathcal{ALCQI}$ to be the underlying DL and thus allowed number restrictions in the witness concept. If we switch the underlying DL to $\mathcal{ALCI}$, then $\mathcal{T}_1 \cup \mathcal{T}_2$ *is* a Γ-conservative extension of $\mathcal{T}_1$, for the same Γ. In the next section, we investigate a purely model-theoretic alternative to the notion above which does not depend on the language.

2.2 Model Conservative Extensions

In mathematical logic and software specification [8], there are two different kinds of conservative extensions: one that is based on the consequence relation "$\models$" as in Definition 1 and one that is based on models only . For simplicity, we formulate this second notion only for the case where $\Gamma = \mathsf{sig}(\mathcal{T}_1)$.

Definition 2 (Model Conservative Extension) *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be TBoxes. We say that $\mathcal{T}_1 \cup \mathcal{T}_2$ is a* model conservative extension *of $\mathcal{T}_1$ iff for every model $\mathcal{I}$ of $\mathcal{T}_1$, there exists a model of $\mathcal{T}_1 \cup \mathcal{T}_2$ which can be obtained from $\mathcal{I}$ by modifying the interpretation of the predicates in $\mathsf{sig}(\mathcal{T}_2) \setminus \mathsf{sig}(\mathcal{T}_1)$ while leaving the predicates in $\mathsf{sig}(\mathcal{T}_1)$ fixed.*

To distinguish the two versions of conservative extensions, in this section we call the original version *deductive conservative extensions.*

The notion of a model conservative extension are more strict than the deduction-based one: if $\mathcal{T}_1 \cup \mathcal{T}_2$ is a model conservative extension of $\mathcal{T}_1$, then it is clearly also a deductive conservative extension of $\mathcal{T}_1$, but the converse does not hold. To see the latter, consider the TBoxes

$$\mathcal{T}_1 = \{\exists r.\top \sqcap \exists s.\top = \top\}, \quad \mathcal{T}_2 = \{\exists r.A \sqcap \exists s.\neg A = \top\}.$$

It is not hard to see that $\mathcal{T}_1 \cup \mathcal{T}_2$ is a deductive conservative extension of $\mathcal{T}_1$ if $\mathcal{ALC}$ (or even $\mathcal{ALCQI}$) is the language for witness concepts, but it is not a model conservative extension. Note that model conservative extensions do not depend on the underlying DL at all: there are no witness concepts on whose language the result of being a model conservative extension could depend.

However, model conservative extensions are a problematic choice: we show that they are highly undecidable even in the basic description logic $\mathcal{ALC}$ (and therefore in all its extensions).

Theorem 3 *It is Π^1_1-hard to decide whether for two given $\mathcal{ALC}$ TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$, $\mathcal{T}_1 \cup \mathcal{T}_2$ is a model conservative extension of $\mathcal{T}_1$.*

Proof. The proof consists of a reduction of the highly undecidable *semantic consequence problem* from modal logic. Before formulating the DL version of this problem, we need some preliminaries. We use $\mathcal{ALC}$ concepts with only one fixed role name r, which we call $\mathcal{ALC}_r$-concepts. Accordingly, we also assume that interpretations interpret only concept names and the role name r. A *frame* is a structure $\mathcal{F} = (\Delta^{\mathcal{F}}, r^{\mathcal{F}})$ where $\Delta^{\mathcal{F}}$ is a non-empty set and $r^{\mathcal{F}} \subseteq \Delta^{\mathcal{F}} \times \Delta^{\mathcal{F}}$. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is *based* on a frame $\mathcal{F}$ iff $\Delta^{\mathcal{I}} = \Delta^{\mathcal{F}}$ and $r^{\mathcal{I}} = r^{\mathcal{F}}$. We say that a concept C is *valid* on $\mathcal{F}$ (written $\mathcal{F} \models C$) iff $C^{\mathcal{I}} = \Delta^{\mathcal{I}}$ for every interpretation $\mathcal{I}$ based on $\mathcal{F}$.

Definition Let D and E be $\mathcal{ALC}_r$-concepts. We say that E is a *semantic consequence* of D iff for every frame $\mathcal{F} = (\Delta^{\mathcal{F}}, r^{\mathcal{F}})$ such that $\mathcal{F} \models D$, it holds that $\mathcal{F} \models E$.

In [10, 9], it is proved that for $\mathcal{ALC}_r$-concepts D and E, the problem "Is E a semantic consequence of D?" is Π^1_1-hard. Clearly, we may assume that in this problem, E and D do not share any atomic concept names. We now show that the semantic consequence problem can be reduced to the model conservativeness problem. For $\mathcal{ALC}_r$-concepts D and E which do not share atomic concept names, we define the TBoxes

$$\mathcal{T}_D = \{\exists u_1.\neg D = \top\}, \mathcal{T}_E = \{\exists u_2.\neg E = \top\},$$

where u_1 and u_2 are distinct role names distinct from r.

Claim. E is a semantic consequence of D if, and only if, $\mathcal{T}_E \cup \mathcal{T}_D$ is a model conservative extension of $\mathcal{T}_E$.

"$\Rightarrow$" We show the contraposition. Assume that E is not a semantic consequence of D. Then there exists a frame $\mathcal{F} = (\Delta^{\mathcal{F}}, r^{\mathcal{F}})$ such that $\mathcal{F} \models D$ and there is an interpretation $\mathcal{I}$ based on $\mathcal{F}$ such that $E^{\mathcal{I}} \neq \Delta^{\mathcal{I}}$. Let $x \in \Delta^{\mathcal{I}}$ with $x \notin E^{\mathcal{I}}$. We may assume that $u_2^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \{x\}$. Then $\mathcal{I}$ is a model of $\mathcal{T}_E$. But obviously it cannot be modified (by possibly changing the interpretation of u_1) so as to obtain a model for $\mathcal{T}_D$.

"$\Leftarrow$" Assume that E is a semantic consequence of D. Let $\mathcal{I} \models \mathcal{T}_E$. Then $\mathcal{I}$ is not based on a frame $\mathcal{F} = (\Delta^{\mathcal{F}}, r^{\mathcal{F}})$ validating E. Since E is a semantic consequence of D, $\mathcal{F}$ is not validating D either, and there is an interpretation $\mathcal{I}'$ based on $\mathcal{F}$ such that $D^{\mathcal{I}'} \neq \Delta^{\mathcal{I}'}$. Take $y \in \Delta^{\mathcal{I}'}$ such that $y \notin D^{\mathcal{I}'}$. Since D is an $\mathcal{ALC}_r$- concept, we may assume that $u_1^{\mathcal{I}'} = \Delta^{\mathcal{I}'} \times \{y\}$. Hence $\mathcal{I}' \models \mathcal{T}_D$. Consequently, $\mathcal{T}_D \cup \mathcal{T}_E$ is a model conservative extension of $\mathcal{T}_E$. ❑

We will not consider model conservative extensions any further in this paper. Deductive conservative extensions appear to be the appropriate notion for all the ontology design tasks discussed in the introduction *as long as the underlying DL is fixed.*

3 Decidability in $\mathcal{ALCQI}$

We give a complexity result for deciding conservative extensions in $\mathcal{ALCQI}$. We use $|C|$ to denote the length of a concept C, and $|\mathcal{T}|$ to denote the *size* $\sum_{C \sqsubseteq D \in \mathcal{T}}(|C| + |D|)$ of a TBox $\mathcal{T}$.

Theorem 4 *It is 2-*ExpTime*-complete to decide conservative extensions in $\mathcal{ALCQI}$. In the case that $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$, there exists a witness concept C of length at most 3-exponential in $|\mathcal{T}_1 \cup \mathcal{T}_2|$ that can be computed in time polynomial in $|C|$. This bound is optimal.*

The lower bounds can be proved exactly in the same way as the 2-ExpTime lower bound for conservative extensions in $\mathcal{ALC}$ [4]. However, the lower bounds do *not* simply transfer from $\mathcal{ALC}$ to $\mathcal{ALCQI}$ since the former is a fragment of the latter. Therefore, it is necessary to go through the proof in [4] and check that it also works for the case of $\mathcal{ALCQI}$. In the following, we concentrate on proving the upper bound. It is established by devising a 2-ExpTime algorithm that, for convenience, decides *non*-conservative extensions.

We start by reminding that $\mathcal{ALCQI}$ has the tree model property. More precisely, a *tree interpretation* is an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, <^{\mathcal{I}})$ equipped with an additional relation $<^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ such that (i) $(\Delta^{\mathcal{I}}, <^{\mathcal{I}})$ is a tree, (ii) $\bigcup_{r \in \mathsf{N_R} \cup \mathsf{N_R}^-} r^{\mathcal{I}} = < \cup <^{-1}$, and (iii) $s^{\mathcal{I}}$ and $r^{\mathcal{I}}$ are disjoint for all distinct roles s and r. In $\mathcal{ALCQI}$, every concept C that is satisfiable w.r.t. a TBox $\mathcal{T}$ is satisfiable in a *tree model* of $\mathcal{T}$, i.e., a model of $\mathcal{T}$ that is a tree interpretation [7]. In this section, when talking of an interpretation or model of a TBox we always mean a tree interpretation.

To develop the algorithm for deciding non-conservative extensions in $\mathcal{ALCQI}$, we introduce a new kind of witness for non-conservativity. The new witnesses are very similar to finite tree interpretations and easier to work with than witness concepts. For a signature Γ, let a *literal type* for Γ be a subset of $\mathsf{lit}(\Gamma) := \{A, \neg A \mid A \in \Gamma \cap \mathsf{N_C}\}$ such that for each $A \in \Gamma \cap \mathsf{N_C}$, $A \in S$ iff $\neg A \notin S$. A Γ-*role* is a role r such that r or r^- is in Γ.

Definition 5 (Γ-tree) *A* Γ-tree $\mathfrak{T} = (W, <, L, O)$ *is a finite intransitive tree* $(W, <)$ *such that each node* $w \in W$ *is labeled by a literal type* $L(w)$ *for* Γ*, each edge* (w, w') *is labeled by a* Γ*-role* $L(w, w')$*, and* $O \subseteq W$ *is a set of leafs of* $(W, <)$*.*

Essentially, a Γ-tree is a finite tree interpretation equipped with an additional unary predicate O denoting a subset of the leafs. The following definition provides a way to relate Γ-trees and actual interpretations.

Definition 6 (Γ-embedding) *Let* $\mathfrak{T} = (W, <, L, O)$ *be a* Γ*-tree with root* $w \in W$*, and* $\mathcal{I}$ *an interpretation with root* $d \in \Delta^{\mathcal{I}}$*. A* Γ*-embedding* $f : \mathfrak{T} \to \mathcal{I}$ *is an injection from* W *to* $\Delta^{\mathcal{I}}$ *such that*

- $f(w) = d$,
- $L(v, v') = r$ *iff* $f(v) r^{\mathcal{I}} f(v')$*, for all* $v, v' \in W$ *and* Γ*-roles* r,
- $C \in L(v)$ *iff* $f(v) \in C^{\mathcal{I}}$*, for all* $v \in W$ *and* $C \in \mathsf{lit}(\Gamma)$,
- *if* $v \notin O$*, then there does not exist a* $d' \in \Delta^{\mathcal{I}} \setminus \mathsf{ran}(f)$ *and a* Γ*-role* r *such that* $f(v) r^{\mathcal{I}} d'$*.*

$\mathfrak{T}$ *is called* Γ-embeddable into $\mathcal{I}$ *if there is a* Γ*-embedding* $f : \mathfrak{T} \to \mathcal{I}$*.*

The definition illustrates that Γ-trees represent a (finite) initial part of (potentially infinite) tree interpretations. This explains the predicate O of Γ-trees: O marks those leafs in the Γ-tree that are not necessarily leafs in the tree interpretation $\mathcal{I}$ that we embed into. We can now establish Γ-trees as witnesses for non-conservativity.

Lemma 7 $\mathcal{T}_1 \cup \mathcal{T}_2$ *is not a conservative extension of* $\mathcal{T}_1$ *w.r.t.* Γ *iff there exists a* Γ*-tree* $\mathfrak{T} = (W, <, L, O)$ *which is* Γ*-embeddable into a model of* $\mathcal{T}_1$ *but not into any model of* $\mathcal{T}_1 \cup \mathcal{T}_2$*.*

Proof. ($\Rightarrow$) Let C be a concept in $\mathcal{ALCQI}(\Gamma)$ which is satisfiable relative to $\mathcal{T}_1$ but not relative to $\mathcal{T}_1 \cup \mathcal{T}_2$. Take a tree model $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I}, <^\mathcal{I})$ of $\mathcal{T}_1$ such that $w \in C^\mathcal{I}$ for its root w. We may assume that $\mathcal{I}$ has finite outdegree [11]. Now let n be the depth $d(C)$ of C, that is, the maximal number of nestings of at-most number restrictions in C.

Let W denote the set of $u \in \Delta^\mathcal{I}$ such that there is a sequence

$$w = w_0 r_1^\mathcal{I} w_1 r_2^\mathcal{I} \cdots r_k^\mathcal{I} w_k = u$$

of length not exceeding n such that each r_i is a Γ-role, for $i \leq k$. Define the Γ-tree $\mathfrak{T} = (W, <, L, O)$ by setting $< = <^\mathcal{I} \cap (W \times W)$, $L(u) = \{D \in \mathsf{lit}(\Gamma) \mid u \in D^\mathcal{I}\}$, defining $L(u, u')$ as the unique Γ-role r such that $u r^\mathcal{I} u'$, and defining O as the set of leafs u of $(W, <)$ such that there exist $u' \notin W$ and a Γ-role r with $u r^\mathcal{I} u'$. It is readily checked that the identity mapping is a Γ-embedding from $\mathfrak{T}$ into $\mathcal{I}$. Conversely, there is no Γ-embedding from $\mathfrak{T}$ into a tree model $\mathcal{I}'$ of $\mathcal{T}_1 \cup \mathcal{T}_2$ because in any such model one would have $u \in C^{\mathcal{I}'}$ for the root u of $\mathcal{I}'$. This can be proved by induction on the subconcepts of C.

($\Leftarrow$) We postpone the proof to Section 5. ❏

The general idea behind the algorithm is as follows: by Lemma 7, to decide whether $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$, it suffices to decide whether there exists a Γ-tree that is Γ-embeddable into a model of $\mathcal{T}_1$, but not into any model of $\mathcal{T}_1 \cup \mathcal{T}_2$. This is what our algorithm will do. Alas, the algorithm cannot simply try to construct the required Γ-tree because there are cases in which the smallest such tree is 3-exponential in $|\mathcal{T}_1 \cup \mathcal{T}_2|$ and we are aming at a 2-ExpTime algorithm. Therefore, we check the existence of the Γ-tree by considering certain witnesses for the existence of such a tree. Before we can introduce these witnesses (which should not be confused with Γ-trees as witnesses for non-conservativity), we need to introduce the notion of a type.

Definition 8 (Type) *Let $\mathcal{T}$ be a TBox. We use $\mathsf{cl}(\mathcal{T})$ to denote the smallest set that contains all concepts in $\mathcal{T}$ and is closed under single negations and under subconcepts. A* $\mathcal{T}$-type *is a subset of $\mathsf{cl}(\mathcal{T})$ such that*

- *$\neg C \in t$ iff $C \notin t$, for all $\neg C \in \mathsf{cl}(\mathcal{T})$;*
- *$C_1 \sqcap C_2 \in t$ iff $C_1 \in t$ and $C_2 \in t$, for all $C_1 \sqcap C_2 \in \mathsf{cl}(\mathcal{T})$.*

Given an interpretation $\mathcal{I}$ and $u \in \Delta^\mathcal{I}$, the set

$$t_\mathcal{I}^\mathcal{T}(u) = \{C \in \mathsf{cl}(\mathcal{T}) \mid u \in C^\mathcal{I}\}$$

is a $\mathcal{T}$-type. In what follows, we will not always distinguish between the type t and the conjunction of all members of t. We now introduce a witness for the existence of a Γ-tree that is Γ-embeddable into a model of $\mathcal{T}_1$, but not into any model of $\mathcal{T}_1 \cup \mathcal{T}_2$. To avoid heavy sub- and superscripting, from now on we assume the input $\mathcal{T}_1$, $\mathcal{T}_2$, and Γ to be fixed.

Definition 9 (Root pair, Internal pair) *A* root pair (t, U) *consists of a $\mathcal{T}_1$-type t and a set U of $\mathcal{T}_1 \cup \mathcal{T}_2$-types. An* internal pair $(t' \rightarrow_r t, U)$ *consists of a Γ-role r, $\mathcal{T}_1$-types t' and t, and a function U mapping each $\mathcal{T}_1 \cup \mathcal{T}_2$-type to a set of $\mathcal{T}_1 \cup \mathcal{T}_2$-types.*

Intuitively, each (root or internal) pair encodes relevant information about possible embeddings of a Γ-tree into models of $\mathcal{T}_1$ and $\mathcal{T}_1 \cup \mathcal{T}_2$. This is made precise by the notion of realizability.

Definition 10 (Realizable root pair) *Let $\mathfrak{T} = (W, <, L, O)$ be a Γ-tree. A root pair (t, U) is* realized *by $\mathfrak{T}$ iff*

1. *there exist a model $\mathcal{I}$ of $\mathcal{T}_1$ with root $d \in t^{\mathcal{I}}$ and a Γ-embedding $f : \mathfrak{T} \rightarrow \mathcal{I}$;*
2. *for every $\mathcal{T}_1 \cup \mathcal{T}_2$-type s, we have $s \in U$ iff there exist a model $\mathcal{I}$ of $\mathcal{T}_1 \cup \mathcal{T}_2$ with root $d \in s^{\mathcal{I}}$ and a Γ-embedding $f : \mathfrak{T} \rightarrow \mathcal{I}$.*

While root pairs encode information about possible embeddings of a Γ-tree into models of $\mathcal{T}_1$ and $\mathcal{T}_1 \cup \mathcal{T}_2$, internal pairs encode information about possible embeddings of a Γ-tree into *rooted submodels* of models of $\mathcal{T}_1$ and $\mathcal{T}_1 \cup \mathcal{T}_2$. In the following, if $\mathcal{I}$ is a (tree) interpretation and $d \in \Delta^{\mathcal{I}}$, we write $\mathcal{I}_d$ to denote the sub-tree interpretation of $\mathcal{I}$ rooted at d.

Definition 11 (Realizable internal pair) *Let $\mathfrak{T} = (W, <, L, O)$ be a Γ-tree. An internal pair $(t' \rightarrow_r t, U)$ is realized by $\mathfrak{T}$ iff*

- *there exist a model $\mathcal{I}$ of $\mathcal{T}_1$ and $d', d \in \Delta^{\mathcal{I}}$ such that $d' \in (t')^{\mathcal{I}}$, $d' r^{\mathcal{I}} d$, $d \in t^{\mathcal{I}}$, and there is a Γ-embedding $f : \mathfrak{T} \rightarrow \mathcal{I}_d$;*
- *for all $\mathcal{T}_1 \cup \mathcal{T}_2$-types s, s', we have $s' \in U(s)$ iff there exist a model $\mathcal{I}$ of $\mathcal{T}_1 \cup \mathcal{T}_2$ and $d', d \in \Delta^{\mathcal{I}}$ such that $d' \in (s')^{\mathcal{I}}$, $d' r^{\mathcal{I}} d$, $d \in s^{\mathcal{I}}$, and there is a Γ-embedding $f : \mathfrak{T} \rightarrow \mathcal{I}_d$.*

A (root or internal) pair is realizable *if there exists a Γ-tree $\mathfrak{T}$ which realizes it.*

Observe that internal pairs store information not only about the element $d \in \Delta^{\mathcal{I}}$ to which the root of $\mathfrak{T}$ is mapped, but also comprise the type t' of the predecessor d' of d in $\mathcal{I}$ and the (unique!) role r which connects d' and d. This is necessary due to the presence of inverse roles and number restrictions and bears some similarity to the *double blocking* technique in tableau algorithms, see [7]. Also note that the U-component of internal pairs is a function rather than a set because, intuitively, the possible types of d in models of $\mathcal{T}_1 \cup \mathcal{T}_2$ depend on the type of the predecessor d' in such models.

Let us now describe the algorithm. By Lemma 7 and definition of realizability, there exists a realizable root pair of the form $(t, \emptyset)$ iff $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$ w.r.t. Γ. The algorithm for deciding non-conservative extensions searches for such a root pair. The easiest case is that a root pair $(t, \emptyset)$ is realized by a *singleton* Γ*-tree*, i.e., a Γ-tree that consists of only a single node. This special case is tested first. If the test is not successful, we must check whether there is a root pair $(t, \emptyset)$ that is realized by a non-singleton tree $\mathfrak{T} = (W, <, L, O)$. Assume that this is the case and that the root of $\mathfrak{T}$ is w. Then each subtree of $\mathfrak{T}$ rooted at a node $w' \in W$ with $w < w'$ realizes an internal pair $(\hat{t}' \rightarrow_{\hat{r}} \hat{t}, \hat{U})$ with $\hat{t}' = t$ and $\hat{r} = L(w, w')$. Intuitively, this means that we can check realization of the root pair $(t, \emptyset)$ in $\mathfrak{T}$ based on the realization of internal pairs in trees of strictly smaller height. Similarly, we can check the realizability of *internal* pairs in a Γ-tree based on the realizability of internal pairs in Γ-trees of strictly smaller height. Based on these observations, our algorithm repeatedly generates internal pairs that are realized by Γ-trees of larger and larger height until all such pairs are generated. It then checks whether there exists a root pair $(t, \emptyset)$ that is realizable based on the generated internal pairs. The following definition formalizes one step of the algorithm in which root pairs or new internal pairs are generated from an existing set of internal pairs.

In the following, if $\mathfrak{T}$ is a Γ-tree and $w \in W$, we write $\mathfrak{T}_w$ to denote the sub-tree of $\mathfrak{T}$ rooted at w.

Definition 12 (One step) *Let $\mathcal{R}$ be a set of internal pairs. A root pair (t, U) (resp. internal pair $(t' \rightarrow_r t, U)$) can be* obtained in one step *from $\mathcal{R}$ if there exists a Γ-tree $\mathfrak{T} = (W, <, L, O)$ with root w such that*

- *$\mathfrak{T}$ realizes (t, U) (resp. $(t' \rightarrow_r t, U)$);*
- *for all $w' \in W$ with $w < w'$, there exists an internal pair $p = (\hat{t}' \rightarrow_{\hat{r}} \hat{t}, \hat{U}) \in \mathcal{R}$ such that $\hat{t}' = t$, $\hat{r} = L(w, w')$, and p is realized by $\mathfrak{T}_{w'}$.*

The details of our algorithm are given in Figure 2, where

$$m_{\mathcal{T}_1, \mathcal{T}_2} := 2 \times |\mathcal{T}_1 \cup \mathcal{T}_2| \times 2^{3 \times |\mathcal{T}_1 \cup \mathcal{T}_2|}.$$

Intuitively, considering only a subset of $\mathcal{R}_i$ of cardinality $m_{\mathcal{T}_1, \mathcal{T}_2}$ means that we limit our attention to Γ-trees of out-degree $m_{\mathcal{T}_1, \mathcal{T}_2}$. This is justified by the following lemma.

Lemma 13 *If $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$ w.r.t. Γ, then there exists a root pair $(t, \emptyset)$ realized by a Γ-tree $\mathfrak{T}$ of outdegree at most $m_{\mathcal{T}_1, \mathcal{T}_2}$.*

Suppose TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$, and a signature $\Gamma \subseteq sig(\mathcal{T}_1)$ are given.

1. Determine the set $\mathcal{R}_0^r$ of root pairs realized by singleton Γ-trees. If $\mathcal{R}_0^r = \emptyset$, then reject the input (because $\mathcal{T}_1$ is not satisfied in any model). If $\mathcal{R}_0^r$ contains a root pair (t, U) such that $U = \emptyset$, then accept. Else,

2. Determine the set $\mathcal{R}_0$ of internal pairs realizable by singleton Γ-trees. If $\mathcal{R}_0 = \emptyset$, then reject the input. Else,

3. Generate the sequence $\mathcal{R}_1, \mathcal{R}_2, \ldots$ of sets of internal pairs such that

$$\mathcal{R}_{i+1} = \mathcal{R}_i \cup \mathcal{R}_i',$$

where $\mathcal{R}_i'$ is the set internal pairs which can be obtained from some non-empty subset of $\mathcal{R}_i$ of cardinality not exceeding $m_{\mathcal{T}_1,\mathcal{T}_2}$ in one step. This is done until $\mathcal{R}_i = \mathcal{R}_i \cup \mathcal{R}_i'$. Then accept the input if there exists a root pair (t, U) with $U = \emptyset$ which can be obtained in one step from some subset of $\mathcal{R}_i$ of cardinality not exceeding $m_{\mathcal{T}_1,\mathcal{T}_2}$. If no such root pair exists, reject the input.

Figure 2: Algorithm for non-conservativeness w.r.t. Γ.

Proof. Suppose $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a Γ-conservative extension of $\mathcal{T}_1$. By Lemma 7, there exists a Γ-tree $\mathfrak{T} = (W, <, L, O)$ which is Γ-embeddable into a model of $\mathcal{T}_1$ but not Γ-embeddable into any model of $\mathcal{T}_1 \cup \mathcal{T}_2$. Consider such a $\mathfrak{T}$ with root w and take a tree-model $\mathcal{I}$ of $\mathcal{T}_1$ with root u and a Γ-embedding $f : \mathfrak{T} \to \mathcal{I}$. Take t to be the $\mathcal{T}_1$-type $t_{\mathcal{I}}^{\mathcal{T}_1}(u)$. Then we have that the root pair (t, U) with $U = \emptyset$ is realized by $\mathfrak{T}$. By recursively removing certain subtrees from $\mathfrak{T}$, we will obtain a Γ-tree satisfying the bound $m_{\mathcal{T}_1,\mathcal{T}_2}$ on the outdegree stated in the lemma and still realizing (t, U). To this end, we require the following claim.

Claim 14 *Let $\mathfrak{T}_v$ be a Γ-tree based on $(W_v, <_v)$ with root v which realizes (a) the root pair (t_0, U_0), or (b) the internal pair $(t_0' \to_r t_0, U_0)$. By possibly removing from $\mathfrak{T}_v$ some nodes from $\{v' \mid v <_v v'\}$ together with the trees $\mathfrak{T}_{v'}$ they generate one can obtain a Γ-tree $\mathfrak{T}_v'$ based on $W_v' \subseteq W_v$ with $\mathfrak{T}_v'$ realizing (a) a root pair (t_0, U_0') with $U_0' \subseteq U_0$, or (b) an internal pair $(t_0' \to_r t_0, U_0')$ with $U_0'(s) \subseteq U_0(s)$, for all $\mathcal{T}_1 \cup \mathcal{T}_2$-types $s \in \mathsf{dom}(U_0')$, such that $|\{v' \in W_v' \mid v <_v v'\}| \leq m_{\mathcal{T}_1,\mathcal{T}_2}$.*

The claim will be proved below. First, however, we show the following Property $(*)$.

$(*)$ Suppose v is a node in $\mathfrak{T}$ and let $t_0 = t_{\mathcal{I}}^{\mathcal{T}_1}(f(v))$. Define $\mathfrak{T}'$ as follows:

- if $v = w$, then let $\mathfrak{T}' = \mathfrak{T}'_v$, where $\mathfrak{T}'_v$ is obtained from $\mathfrak{T}_v$ as in Case (a) of Claim 14;

- if $v \neq w$, then let v' be the r^--predecessor of v in $\mathfrak{T}$, i.e., $v' < v$ and $L(v', v) = r$. Set $t'_0 = t^{\mathcal{T}_1}_{\mathcal{I}}(f(v'))$. In this case assume that $\mathfrak{T}'_v$ is obtained from $\mathfrak{T}_v$ as in Case (b) of Claim 14 and that $\mathfrak{T}' = (W', <', L', O')$ is obtained from $\mathfrak{T}$ by replacing its subtree $\mathfrak{T}_v$ by $\mathfrak{T}'_v$.

Then $\mathfrak{T}'$ still realizes (t, U) with $U = \emptyset$.

The Property $(*)$ clearly holds in the case where $v = w$. Suppose now that $v \neq w$. We consider the second component $U = \emptyset$ of the root pair (t, U) only. For the first component, t, the proof is simpler and left to the reader. Let $v = v_0, v_1, \ldots, v_n = w$ be the sequence of nodes such that, for all $i \leq n$, $L'(v_{i+1}, v_i) = r_i$, for some Γ-roles r_i. Let U_i and U'_i be the second component of the pairs realized by $\mathfrak{T}_{v_i}$ and $\mathfrak{T}'_{v_i}$, respectively.

We show by induction on $i \geq 0$ that (1) if $v_i = w$, then $U'_i \subseteq U_i$, and (2) if $v_i \neq w$, then $U'_i(s) \subseteq U_i(s)$, for each $\mathcal{T}_1 \cup \mathcal{T}_2$-type s. For the induction base, where $i = 0$, the properties (1) and (2) directly follow from the items (a) and (b) of Claim 14. For the induction step, assume (1) and (2) were already shown for i. Here we only show Property (2) for $i + 1$; (1) can be shown similarly. Suppose $s'' \in U'_{i+1}(s')$, for two $\mathcal{T}_1 \cup \mathcal{T}_2$-types s', s''. We have to show $s'' \in U_{i+1}(s')$. By Definition 11 for realizable internal pairs, there exists a tree model $\mathcal{I}'$ of $\mathcal{T}_1 \cup \mathcal{T}_2$ with nodes u'_1, u'_2 such that $u'_1 \in (s')^{\mathcal{I}'}$, $u'_2 \in (s'')^{\mathcal{I}'}$, $u'_2 (r_{i+1})^{\mathcal{I}'} u'_1$, and a Γ-embedding

$$f' : \mathfrak{T}'_{v_{i+1}} \to \mathcal{I}'_{u'_1}.$$

Let $s = t^{\mathcal{T}_1 \cup \mathcal{T}_2}_{\mathcal{I}'}(u'_0)$ be the $\mathcal{T}_1 \cup \mathcal{T}_2$-type determined by the node $u'_0 = f'(v_i)$ in $\mathcal{I}'$. See Figure 3 for an illustration of the Γ-tree $\mathfrak{T}'$, the tree-model $\mathcal{I}'$, and the Γ-embedding f'. By the induction hypothesis, we have $s' \in U_i(s)$. Then, again by Definition 11, there exists a tree-model $\mathcal{I}$ of $\mathcal{T}_1 \cup \mathcal{T}_2$ with nodes u_0, u_1 such that $u_0 \in s^{\mathcal{I}}$, $u_1 \in (s')^{\mathcal{I}}$, $u_1 r_i^{\mathcal{I}} u_0$, and a Γ-embedding

$$f : \mathfrak{T}_{v_i} \to \mathcal{I}_{u_0}.$$

To show $s'' \in U_{i+1}(s')$, we construct a tree-model $\mathcal{I}''$ of $\mathcal{T}_1 \cup \mathcal{T}_2$ and a Γ-embedding

$$f'' : \mathfrak{T}_{v_{i+1}} \to \mathcal{I}''_{u'_1}.$$

Obtain $\mathcal{I}''$ from $\mathcal{I}'$ by replacing the submodel generated by u'_0 in $\mathcal{I}'$ by the submodel generated by u_0 in $\mathcal{I}$. Construct a Γ-embedding $f'' : \mathfrak{T}_{v_{i+1}} \to \mathcal{I}''_{u'_1}$ using f and f' as follows: for each node u in $\mathfrak{T}_{v_{i+1}}$, set

$$f''(u) = \begin{cases} f(u) & \text{if } u = v_i \text{ or } v_i \text{ is an ancestor of } u \text{ in } \mathfrak{T}_{v_{i+1}}; \\ f'(u) & \text{otherwise.} \end{cases}$$

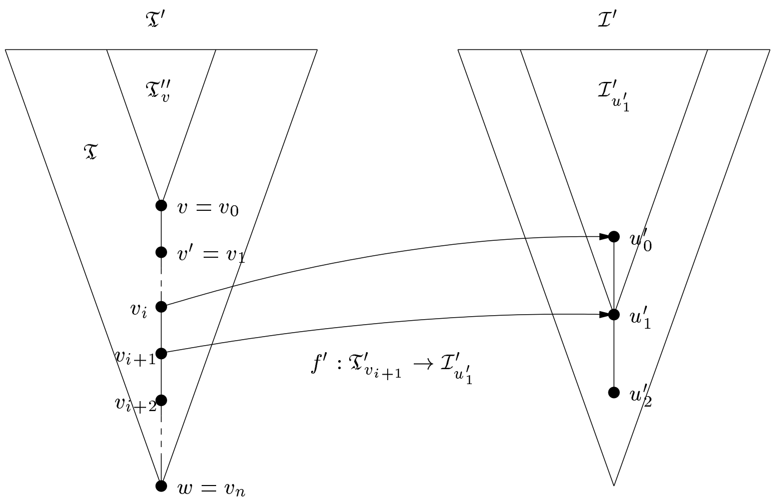

Figure 3: Γ-tree $\mathfrak{T}'$, tree-model $\mathcal{I}'$, and Γ-embedding f'.

See Figure 4 for an illustration of the Γ-tree $\mathfrak{T}$, the tree-model $\mathcal{I}''$, and the Γ-embedding f''. To see that $\mathcal{I}''$ is indeed a model of $\mathcal{T}_1 \cup \mathcal{T}_2$, it suffices to show the following property: for all $C \in \mathsf{cl}(\mathcal{T}_1 \cup \mathcal{T}_2)$ and all $x \in \Delta^{\mathcal{I}''}$:

- if $x \in \Delta^{\mathcal{I}}$, then $x \in C^{\mathcal{I}}$ iff $x \in C^{\mathcal{I}''}$;
- if $x \in \Delta^{\mathcal{I}'}$, then $x \in C^{\mathcal{I}'}$ iff $x \in C^{\mathcal{I}''}$.

Using the construction of $\mathcal{I}''$ and the fact that

$$t_{\mathcal{I}'}^{\mathcal{T}_1 \cup \mathcal{T}_2}(u'_0) = s = t_{\mathcal{I}}^{\mathcal{T}_1 \cup \mathcal{T}_2}(u_0)$$

and

$$t_{\mathcal{I}'}^{\mathcal{T}_1 \cup \mathcal{T}_2}(u'_1) = s' = t_{\mathcal{I}}^{\mathcal{T}_1 \cup \mathcal{T}_2}(u_1),$$

this can be shown by an induction on the structure of the concepts C. Moreover, it is readily checked that the nodes u'_1, u'_2 in $\mathcal{I}''$ are such that $u'_1 \in (s')^{\mathcal{I}''}$, $u'_2 \in (s'')^{\mathcal{I}''}$, $u'_2 (r_{i+1})^{\mathcal{I}''} u'_1$, and that f'' is indeed a Γ-embedding. This finishes the induction step and $(*)$ is proved.

Hence, the root pair (t, U) (with $U = \emptyset$) is realized by $\mathfrak{T}'$. Thus we can, starting from $\mathfrak{T}$ successively remove generated subtrees of nodes with more than $m_{\mathcal{T}_1,\mathcal{T}_2}$ successors and obtain a Γ-tree $\mathfrak{T}'$ whose outdegree is bounded by $m_{\mathcal{T}_1,\mathcal{T}_2}$ such that $(t, \emptyset)$ is realized by $\mathfrak{T}'$.

Finally, we state again and prove the claim used above.

Claim 15 *Let $\mathfrak{T} = (W, <, L, O)$ be a Γ-tree with root v which realizes (a) the root pair (t_0, U_0), or (b) the internal pair $(t'_0 \to_r t_0, U_0)$. By possibly removing from $\mathfrak{T}$ some nodes from $\{v' \mid v < v'\}$ together with the trees $\mathfrak{T}_{v'}$*

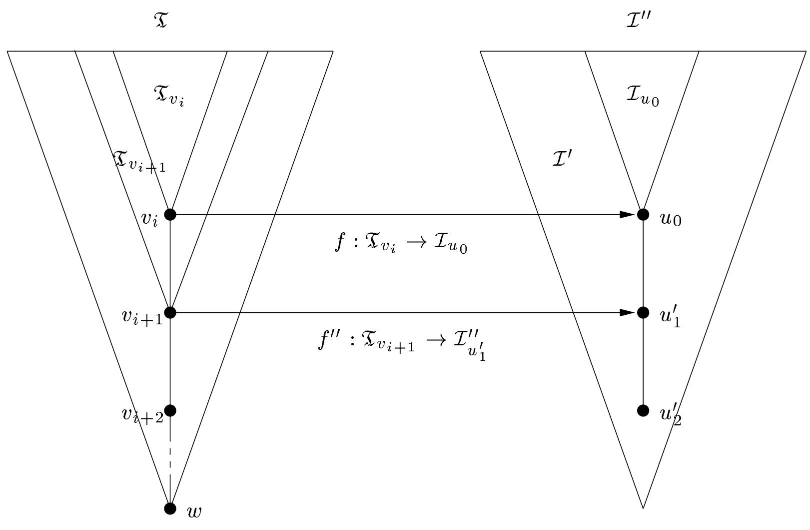

Figure 4: Γ-tree $\mathfrak{T}$, tree-model $\mathcal{I}''$, and Γ-embedding f''.

they generate one can obtain a Γ*-tree* $\mathfrak{T}' = (W', <', L', O')$ *with* $\mathfrak{T}'$ *realizing (a) a root pair* (t_0, U'_0) *with* $U'_0 \subseteq U_0$*, or (b) an internal pair* $(t'_0 \to_r t_0, U'_0)$ *with* $U'_0(s) \subseteq U_0(s)$*, for all* $\mathcal{T}_1 \cup \mathcal{T}_2$*-types* $s \in \mathsf{dom}(U'_0)$*, such that* $|\{v' \mid v <' v'\}| \leq m_{\mathcal{T}_1,\mathcal{T}_2}$.

Proof. Let $\mathfrak{T}$ be as in the claim. Here we consider Case (b) where $(t' \to_r t, U)$ is realized by $\mathfrak{T}$ only. The proof for Case (a) is similar and left to the reader. Let $W_v = \{v' \mid v < v'\}$ be the set of successor nodes of v in $\mathfrak{T}$. We define a set $W' \subseteq W_v$ by selecting subsets of W_v in two steps.

First, we select a set $W^1 \subseteq W_v$ of nodes as follows. Since $\mathfrak{T}$ realizes $(t' \to_r t, U)$, we have by Definition 11 for realizable internal pairs that there exists a model $\mathcal{I}$ of $\mathcal{T}_1$ with nodes u, u' such that $u \in t^{\mathcal{I}}$, $u' \in (t')^{\mathcal{I}}$, $u' r^{\mathcal{I}} u$, and a Γ-embedding

$$f : \mathfrak{T} \to \mathcal{I}_u.$$

See Figure 5 for an illustration of the Γ-tree $\mathfrak{T}$, the tree-model $\mathcal{I}$, and the Γ-embedding f. Then we select, for each concept $\neg(\leqslant n\, r'\, C) \in t$,

- if $r' = r^-$ and $C \in t'$: n many r'-successors v' of v in $\mathfrak{T}$ such that $C \in t_{\mathcal{I}}^{\mathcal{T}_1}(f(v'))$; or
- if $r' \neq r^-$ or $C \notin t'$: $n+1$ many r'-successors v' of v in $\mathfrak{T}$ such that $C \in t_{\mathcal{I}}^{\mathcal{T}_1}(f(v'))$.

Let W^1 be the set of all successors of v selected in this way.

In the second step, we select, for each $\mathcal{T}_1 \cup \mathcal{T}_2$-type s and Γ-role r' a set $W^2_{s,r'} \subseteq W_v$ of nodes as follows.

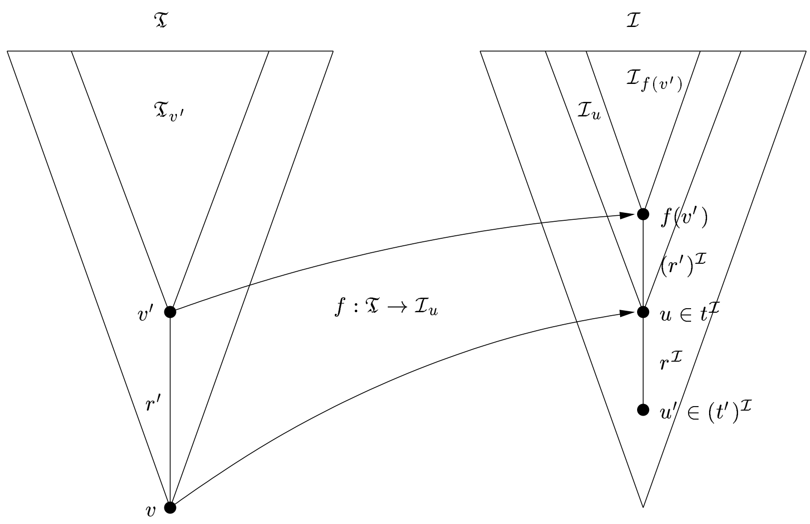

Figure 5: Γ-tree $\mathfrak{T}$, tree-model $\mathcal{I}$, and Γ-embedding f.

For each v' with $L(v, v') = r'$ let $U_{v'}$ be the second component of an internal pair $(t_{v'} \to_{r'} t'_{v'}, U_{v'})$ realized by $\mathfrak{T}_{v'}$. Let S_s be a subset of s of the form

$$S_s = \{(\leqslant n_1\ r'\ C_1), \ldots, (\leqslant n_m\ r'\ C_m)\}.$$

We call an r'-successor v' of v in $\mathfrak{T}$ an *S_s-witness* if for all $\mathcal{T}_1 \cup \mathcal{T}_2$-types s' with $s \in U_{v'}(s')$, it holds

$$\{C_1, \ldots, C_m\} \cap s' \neq \emptyset.$$

Let m_{S_s} be the number of S_s-witnesses. Take $W^2_{s,r'}$ to be the minimal set containing, for all such subsets S_s of s, $\min\{m_{S_s}, n_1 + n_2 + \cdots + n_m + 1\}$ many r'-successor nodes of v that are S_s-witnesses.

Now let W' be the union of W^1 and the sets $W^2_{s,r'}$, for each $\mathcal{T}_1 \cup \mathcal{T}_2$-type s and Γ-role r'. Define a new Γ-tree $\mathfrak{T}' = (W', <', L')$ by removing from $\mathfrak{T}$ all subtrees $\mathfrak{T}_{v'}$ generated by nodes $v' \in W_v$ which where not selected to be in W'.

We analyze the cardinality of W'. Recall that numbers in concepts with number restrictions are coded in binary. The set W^1 is of cardinality at most $|\mathcal{T}_1| \times 2^{|\mathcal{T}_1|}$: in t there are at most $|\mathcal{T}_1|$ many negated at-most number restrictions, for which we each select not more than $2^{|\mathcal{T}_1|}$ many nodes. There are $2^{|\mathcal{T}_1 \cup \mathcal{T}_2|}$ many $\mathcal{T}_1 \cup \mathcal{T}_2$-types s and each set W^2_s has not more than $|\mathcal{T}_1 \cup \mathcal{T}_2| \times 2^{2 \times |\mathcal{T}_1 \cup \mathcal{T}_2|}$ many elements: given s, there are at most $2^{|\mathcal{T}_1 \cup \mathcal{T}_2|}$ many subsets S of s each containing not more than $|\mathcal{T}_1 \cup \mathcal{T}_2|$ many at-most number restrictions, for which we select at most $2^{|\mathcal{T}_1 \cup \mathcal{T}_2|}$ many nodes. Hence, we have

$m_{\mathcal{T}_1,\mathcal{T}_2} = 2 \times |\mathcal{T}_1 \cup \mathcal{T}_2| \times 2^{3\times|\mathcal{T}_1\cup\mathcal{T}_2|}$ as an upper bound for the cardinality of W'.

In the following, we show that an internal pair $(t' \to_r t, U')$ with $U'(s') \subseteq U(s')$, for all $\mathcal{T}_1\cup\mathcal{T}_2$-types $s' \in \mathsf{dom}(U')$, is realized by $\mathfrak{T}'$. First, we show that the first condition in Definition 11 for realizable internal pairs is satisfied. To this end, we construct a model $\mathcal{I}'$ of $\mathcal{T}_1$ and a Γ-embedding

$$f' : \mathfrak{T}' \to \mathcal{I}'_u$$

as required. Construct a model $\mathcal{I}'$ from the model $\mathcal{I}$ introduced above using the Γ-embedding $f : \mathfrak{T} \to \mathcal{I}_u$ by removing the submodels of $\mathcal{I}$ generated by the nodes $f(v')$, for all tree-nodes $v' \in W_v$ that were not selected to be in W'. By construction, $\mathcal{I}'$ still contains the nodes u, u' with $u' r^{\mathcal{I}'} u$. Look at Figure 5 for an illustration: on the left hand side, it shows the Γ-tree $\mathfrak{T}$ containing the subtrees $\mathfrak{T}_{v'}$ that where removed from $\mathfrak{T}$ to obtain $\mathfrak{T}'$ and, respectively, on the other side, we have the tree-model $\mathcal{I}$ containing the submodels $\mathcal{I}_{f(v')}$ that were removed to obtain $\mathcal{I}'$. Obtain a Γ-embedding $f' : \mathfrak{T}' \to \mathcal{I}'_u$ from f by removing from the domain of f all nodes of the subtrees $\mathfrak{T}_{v'}$ generated by the nodes v', for all nodes $v' \in W_v$ that were not selected to be in W'. It is readily checked that f' is indeed a Γ-embedding. To see that $\mathcal{I}'$ is indeed a model of $\mathcal{T}_1$ and that $u \in t^{\mathcal{I}'}$ and $u' \in (t')^{\mathcal{I}'}$, it suffices to show the following property: for all $C \in \mathsf{cl}(\mathcal{T}_1)$ and all $x \in \Delta^{\mathcal{I}'}$:

$$x \in C^{\mathcal{I}} \text{ iff } x \in C^{\mathcal{I}'}.$$

We show this property by an induction on the structure of the concepts C. The cases for concept names and Boolean constructors are left to the reader. Here we only consider the case where $C = (\leqslant n\ r'\ D)$. Moreover, we only consider the case for the node $x = u$ in $\mathcal{I}'$. Note that this is the only interesting case since the models $\mathcal{I}$ and $\mathcal{I}'$ only differ in the successors of the node $x = u$. For the left-to-right direction of the equation, we have that $u \in (\leqslant n\ r'\ D)^{\mathcal{I}}$ implies $u \in (\leqslant n\ r'\ D)^{\mathcal{I}'}$. This is because u has, by construction, not more r'-successors in $\mathcal{I}'$ than in $\mathcal{I}$. We show the other direction by contrapositive. Suppose $u \notin (\leqslant n\ r'\ D)^{\mathcal{I}}$. That is, $u \in \neg(\leqslant n\ r'\ D)^{\mathcal{I}}$. By the fact that $u \in t^{\mathcal{I}}$, we have $\neg(\leqslant n\ r'\ D) \in t$. Then, by construction, we have:

(a) if $r' = r^-$ and $D \in t'$, then W' contains at least n many r'-successors v' of v such that $f(v') \in D^{\mathcal{I}}$, or

(b) if $r' \neq r^-$ or $D \notin t'$, then W' contains at least $n+1$ many r'-successors v' of v such that $f(v') \in D^{\mathcal{I}}$.

Consider Case (a). Since $u' \in (t')^{\mathcal{I}}$ and $D \in t'$, we have $u' \in D^{\mathcal{I}}$, and, by the induction hypothesis, $u' \in D^{\mathcal{I}'}$. Note that $u(r^-)^{\mathcal{I}'} u'$. But then, by the induction hypothesis, we have that there are at least $n+1$ many nodes $u'' \in \Delta^{\mathcal{I}'}$ such that $u(r')^{\mathcal{I}'}u''$ and $u'' \in D^{\mathcal{I}'}$. Hence $u \in \neg(\leqslant n\, r'\, D)^{\mathcal{I}'}$. Case (b) is easier: we simply apply the induction hypothesis and obtain $u \in \neg(\leqslant n\, r'\, D)^{\mathcal{I}'}$. This finishes the induction. Hence, the first condition in Definition 11 for $(t' \to_r t, U')$ being realized by $\mathfrak{T}'$ is satisfied.

Second, we show $U'(s) \subseteq U(s)$, for all $\mathcal{T}_1 \cup \mathcal{T}_2$-types $s \in \mathsf{dom}(U')$, for the internal pair $(t' \to_r t, U')$ realized by $\mathfrak{T}'$. Assume for $\mathcal{T}_1 \cup \mathcal{T}_2$-types s, s': $s' \in U'(s)$. Let v' with $v' \in W_v$ be an r'-successor of v that was not selected to be in W'. Let $\mathfrak{T}''$ be the Γ-tree obtained from $\mathfrak{T}'$ by adding the subtree $\mathfrak{T}_{v'}$ generated by v' in $\mathfrak{T}$. Let V be the second component of the internal pair $(t' \to_r t, V)$ realized by $\mathfrak{T}''$. We show that $s' \in V(s)$. By subsequently repeating this argument for all such v', we can conclude $U'(s) \subseteq U(s)$.

In the following, we show that $s' \in V(s)$. Since $(t' \to_r t, U')$ with $s' \in U'(s)$ is realized by $\mathfrak{T}'$, we have by Definition 11 for realizable internal pairs that there is a model $\mathcal{I}'$ of $\mathcal{T}_1 \cup \mathcal{T}_2$ with nodes u, u' such that $u \in s^{\mathcal{I}'}$, $u' \in (s')^{\mathcal{I}'}$, $u' r^{\mathcal{I}'} u$, and a Γ-embedding

$$f' : \mathfrak{T}' \to \mathcal{I}'_u.$$

To show that $s' \in V(s)$, we construct a tree-model $\mathcal{I}''$ of $\mathcal{T}_1 \cup \mathcal{T}_2$ and a Γ-embedding

$$f'' : \mathfrak{T}'' \to \mathcal{I}''_u.$$

We first show that the r'-successor v' of v is not an S_s-witness, for any subset S_s of s of the form

$$S_s = \{(\leqslant n_1\, r'\, C_1), \ldots, (\leqslant n_m\, r'\, C_m)\}.$$

For suppose otherwise. Then consider such a set S_s for which v' is an S_s-witness. Then, for every $\mathcal{T}_1 \cup \mathcal{T}_2$-type s'' with $s \in U_{v'}(s'')$, we have

$$\{C_1, \ldots, C_m\} \cap s'' \neq \emptyset.$$

By the fact $v' \notin W'$, we have that $v' \notin W_{s,r'}$. Then it follows from the minimality condition in the definition of $W_{s,r'}$ that $m_{S_s} > n_1 + n_2 + \cdots + n_m + 1$. Hence, W' contains at least $n_1 + n_2 + \cdots + n_m + 1$ many r'-successor nodes v'' of v that are S_s-witnesses. But then, for each such node v'', we have the following. Let $s_{v''} = t_{\mathcal{I}'}^{\mathcal{T}_1 \cup \mathcal{T}_2}(u'')$ be the $\mathcal{T}_1 \cup \mathcal{T}_2$-type determined by the node $u'' = f'(v'')$ in $\mathcal{I}'$. But then, by the fact that v'' is an S_s-witness, we have $u'' \in C_i^{\mathcal{I}'}$, for some i with $1 \leq i \leq m$. Since there are at least $n_1 + n_2 + \cdots + n_m + 1$ such nodes v'', the conjunction of the at-most number restrictions in S_s cannot be satisfied at u in $\mathcal{I}'$. Then, since $S_s \subseteq s$, we have $u \notin s^{\mathcal{I}'}$; a contradiction.

Now, as v' is not an S_s-witness for any such subset S_s of s, we find a $\mathcal{T}_1 \cup \mathcal{T}_2$-type s'' with $s \in U_{v'}(s'')$ such that

$$\{C \mid \exists n \text{ s.t. } (\leqslant n\, r'\, C) \in s\} \cap s'' = \emptyset.$$

Take a model $\mathcal{I}$ of $\mathcal{T}_1 \cup \mathcal{T}_2$ such that $u_0 \in (s'')^{\mathcal{I}}$, $u \in s^{\mathcal{I}}$, $u\,(r')^{\mathcal{I}} u_0$ and there is a Γ-embedding $f : \mathfrak{T}_{v'} \to \mathcal{I}_{u_0}$.

Obtain $\mathcal{I}''$ from the model $\mathcal{I}'$ and $\mathcal{I}$ introduced above by adding the submodel $\mathcal{I}_{u_0}$ of $\mathcal{I}$ generated by the node u_0 to $\mathcal{I}'$ together with $u\,(r')^{\mathcal{I}''} u_0$. By construction, $\mathcal{I}''$ still contains the nodes u, u' with $u' r^{\mathcal{I}''} u$. Define a Γ-embedding $f'' : \mathfrak{T}'' \to \mathcal{I}''_u$ using f' and f as follows: for each node v'' in $\mathfrak{T}''$,

$$f''(v'') = \begin{cases} f(v'') & \text{if } v'' = v' \text{ or } v' \text{ is an ancestor of } v'' \text{ in } \mathfrak{T}''; \\ f'(v'') & \text{otherwise.} \end{cases}$$

It is readily checked that f'' is indeed a Γ-embedding. See Figure 6 for an illustration of the Γ-tree $\mathfrak{T}''$, the tree-model $\mathcal{I}''$, and the Γ-embedding f''.

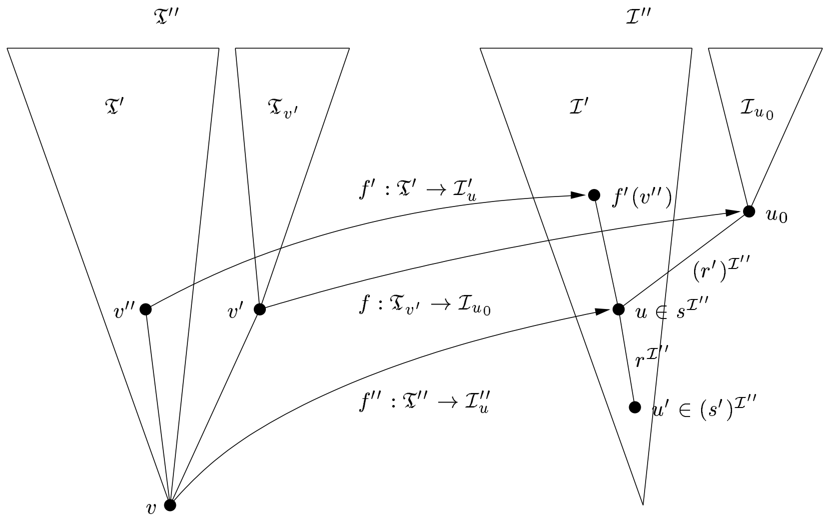

Figure 6: Γ-tree $\mathfrak{T}''$, tree-model $\mathcal{I}''$, and Γ-embedding f''.

To see that $\mathcal{I}''$ is indeed a model of $\mathcal{T}_1 \cup \mathcal{T}_2$, it suffices to show the following property: for all $C \in \mathsf{cl}(\mathcal{T}_1 \cup \mathcal{T}_2)$ and all $x \in \Delta^{\mathcal{I}''}$:

- if $x \in \Delta^{\mathcal{I}}$, then $x \in C^{\mathcal{I}}$ iff $x \in C^{\mathcal{I}''}$;
- if $x \in \Delta^{\mathcal{I}'}$, then $x \in C^{\mathcal{I}'}$ iff $x \in C^{\mathcal{I}''}$.

We show this property by an induction on the structure of the concepts C. The cases for concept names and Boolean constructors are left to the reader. We only consider the case where $C = (\leqslant n\ r'\ D)$. Moreover, we only consider the case for the node $x = u$ in $\mathcal{I}''$. The cases for the other nodes $x \neq u$ are trivial. Since $u \in \Delta^{\mathcal{I}'}$, we have to show

$$u \in (\leqslant n\ r'\ D)^{\mathcal{I}'} \text{ iff } u \in (\leqslant n\ r'\ D)^{\mathcal{I}''}.$$

The direction from right to left holds trivially since, by construction, u has not more r'-successors in $\mathcal{I}'$ than in $\mathcal{I}''$. For the other direction, suppose $u \in (\leqslant n\, r'\, D)^{\mathcal{I}'}$. Notice that we have that v' is not an S_s-witness, for all subsets S_s of s of the form $S_s = \{(\leqslant n_1\, r'\, C_1), \ldots, (\leqslant n_m\, r'\, C_m)\}$ containing the concept $(\leqslant n\, r'\, D)$. Then from the fact that there are at most n many r'-successors of u in $\mathcal{I}'$, it follows by construction that there are also at most n many such r'-successors in $\mathcal{I}''$. Hence, $u \in (\leqslant n\, r'\,)D^{\mathcal{I}''}$. This finishes the induction. We established for $\mathcal{I}''$ that $u \in s^{\mathcal{I}''}$ and $u' \in (s')^{\mathcal{I}''}$. Hence, $s' \in V(s)$. ❑

❑

It remains to be shown that each step of the algorithm can be carried out effectively and that the algorihm yields the 2-ExpTime upper bound stated in Theorem 4. We start with the former. The proof of the following lemma relies on the fact that satisfiability in $\mathcal{ALCQI}$ relative to TBoxes can be decided in ExpTime [11].

Lemma 16 *It can be checked in 2-exponential time (in the size of $\mathcal{T}_1$, $\mathcal{T}_2$) whether a (root or internal) pair can be obtained in one step from a nonempty set $\mathcal{R}$ of realizable internal pairs with $|\mathcal{R}| \leq m_{\mathcal{T}_1,\mathcal{T}_2}$.*

Proof. Let $\mathcal{R}$ be as in the lemma. We only prove the lemma for internal pairs; for root pairs it can be shown similarly. We show that an internal pair $(t' \to_r t, U)$ can be obtained in one step from $\mathcal{R}$ if, and only if, the following holds:

$(*)$ there is a set S with $|S| = n \leq m_{\mathcal{T}_1,\mathcal{T}_2}$ and a function $g : S \to \mathcal{R}$ with $g(v) = (t \to_{r_v} t_v, U_v)$ for $v \in S$ such that the following holds:

(i) $s' \in U(s)$ iff there exist $\mathcal{T}_1 \cup \mathcal{T}_2$-types $s_v \in \mathsf{dom}(U_v)$ such that $s \in \bigcap_{v \in S} U_v(s_v)$ and the conjunction $C_{s,s',r,\overline{s_v}}$ of the following concepts is satisfiable relative to $\mathcal{T}_1 \cup \mathcal{T}_2$:

  - for every Γ-role $r' \neq r^-$,
  $$R \sqcap s \sqcap (= n_{r'}\; r'\; \top),$$
  where $n_{r'} = |\{v \in S \mid r_v = r'\}|$;
  - the concept
  $$R \sqcap s \sqcap (= n_{r'} + 1\; r^-\; \top),$$
  where $n_{r'} = |\{v \in S \mid r_v = r^-\}|$;
  - the concept
  $$\bigsqcap_{v \in S, r_v \neq r^-} (= 0\; r_v\; \neg R) \sqcap (= 1\; r^-\; \neg R \sqcap s'),$$

– for every Γ-role r' and $\mathcal{T}_1 \cup \mathcal{T}_2$-type s''

$$(= m_{r'}\ r'\ R \sqcap s''),$$

where $m_{r'} = |\{v \in S \mid r_v = r' \text{ and } s_v = s''\}|$.

(ii) the above conjunction, where s is replaced by t, s' by t', and the s_v are replaced by the t_v (for $v \in V$), is satisfiable relative to $\mathcal{T}_1$.

It should be clear that $(*)$ holds whenever $(t' \rightarrow_r t, U)$ can be obtained in one step from $\mathcal{R}$. Conversely, suppose that $(*)$ holds. Assume that S and g are chosen in $(*)$. Since the internal pairs $(t \rightarrow_{r_v} t_v, U_v)$ (with $v \in S$) were each realizable, there exists, for each such pair, a Γ-tree $\mathfrak{T}_v$ with a root v such that $(t \rightarrow_{r_v} t_v, U_v)$ is realized by $\mathfrak{T}_v$. We construct a Γ-tree $\mathfrak{T} = (W, <, L, O)$ by taking a new node w (the root of $\mathfrak{T}$) and connecting w to each $\mathfrak{T}_v$, $v \in V$, (assuming the $\mathfrak{T}_v$ are mutually disjoint) by setting $w < v$ and

- $L(w, v) = r_v$,
- $L(w) = \mathsf{lit}(\Gamma) \cap t$ and
- taking O to be the union of the O_v from $\mathfrak{T}_v$.

To show that $(t' \rightarrow_r t, U)$ is obtained in one step from $\mathcal{R}$, it is sufficient to show that $(t' \rightarrow_r t, U)$ is realized by $\mathfrak{T}$. Here we only consider the second condition for internal pairs in Definition 11; the first one can be shown similarly using (ii). Let U' be the second component of the internal pair realized by $\mathfrak{T}$. We have to show $U = U'$. Let s, s' be $\mathcal{T}_1 \cup \mathcal{T}_2$-types and suppose $s' \in U(s)$. Choose s_v, $v \in V$ such that the concept $C_{s,s',r,\overline{s_v}}$ in (i) is satisfiable relative to $\mathcal{T}_1 \cup \mathcal{T}_2$. Let $\mathcal{I}$ be a tree-model of $\mathcal{T}_1 \cup \mathcal{T}_2$ with $d \in C^{\mathcal{I}}_{s,s',r,\overline{s_v}}$. It follows that there are mutually different $f(v)$ in $\Delta^{\mathcal{I}}$, $v \in V$, such that

- $d r_v^{\mathcal{I}} f(v)$ and
- $f(v) \in s_v^{\mathcal{I}}$

and a $d' \in \Delta^{\mathcal{I}}$ different from the $f(v)$, $v \in V$, with $d' \in (s')^{\mathcal{I}}$, and $d' r^{\mathcal{I}} d$. d has no neighbours different from d' and $f(v)$, $v \in V$, in $\mathcal{I}$. We may assume that $d' <^{\mathcal{I}} d$ and $d <^{\mathcal{I}} f(v)$ for all $v \in V$. Moreover, we may assume that there are Γ-embeddings $f_v : \mathfrak{T}_v \rightarrow \mathcal{I}_{f(v)}$. Now we can construct a Γ-embedding $f : \mathfrak{T} \rightarrow \mathcal{I}_d$ by setting $f(u) = f_v(u)$ for u in $\mathfrak{T}_v$ and $f(w) = d$. This Γ-embedding shows that $s' \in U'(s)$.

Conversely, suppose that $s' \in U'(s)$. Then there exists a tree model $\mathcal{I}$ of $\mathcal{T}_1 \cup \mathcal{T}_2$ with $d, d' \in \Delta^{\mathcal{I}}$ such that there is a Γ-embedding $f : \mathfrak{T} \rightarrow \mathcal{I}_d$ and $d \in s^{\mathcal{I}}$, $d' r^{\mathcal{I}} d$, and $d' \in (s')^{\mathcal{I}}$. Let $s_v = t_{\mathcal{I}}^{\mathcal{T}_1 \cup \mathcal{T}_2}(f(v))$, for all $v \in V$. We may assume that $R^{\mathcal{I}} = \{f(v) \mid v \in V \cup \{w\}\}$. It is readily checked that $d \in C^{\mathcal{I}}_{s,s',r,\overline{s_v}}$. Hence $s' \in U(s)$.

Now we analyze the time complexity of deciding $(*)$. Clearly, the number of g, S one has to consider is bounded by $m_{\mathcal{T}_1,\mathcal{T}_2} \times 2^{(m_{\mathcal{T}_1,\mathcal{T}_2})^2}$. For each such g, S we have to check the satisfiability of at most

$$2^{4\times|\mathcal{T}_1\cup\mathcal{T}_2|} \times 2^{m_{\mathcal{T}_1,\mathcal{T}_2}\times 2^{|\mathcal{T}_1\cup\mathcal{T}_2|}}$$

many differenct $\mathcal{ALCQI}$-concepts relative to $\mathcal{T}_1 \cup \mathcal{T}_2$ (res. $\mathcal{T}_1$) of length at most exponential in $|\mathcal{T}_1 \cup \mathcal{T}_2|$. Each satisfiability test can be done in doubly exponential time in $|\mathcal{T}_1 \cup \mathcal{T}_2|$. Hence, the time complexity is at most doubly exponential in $|\mathcal{T}_1 \cup \mathcal{T}_2|$. ❏

Theorem 17 *The algorithm in Figure 2 accepts input $\mathcal{T}_1$, $\mathcal{T}_2$, Γ, if, and only if, $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a Γ-conservative extension of $\mathcal{T}_1$. It runs in 2-exponential time.*

Proof. We prove correctness, completeness and the complexity statement:

- Correctness: Suppose the algorithm accepts $\mathcal{T}_1$, $\mathcal{T}_2$, Γ. There are two cases: in the first case there is a singleton Γ-tree realizing a root pair (t, U) such that $U = \emptyset$ and in the second case there is a non-singleton Γ-tree which realizes a root pair (t, U) with $U = \emptyset$. In both cases $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a Γ-conservative extension of $\mathcal{T}_1$.

- Completeness: Suppose $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a Γ-conservative extension of $\mathcal{T}_1$. By Lemma 14 there exists a Γ-tree $\mathfrak{T}$ with root w which has outdegree not exceeding $m_{\mathcal{T}_1,\mathcal{T}_2}$ such that $\mathfrak{T}$ is Γ-embeddable into a tree model of $\mathcal{T}_1$ but not into any model of $\mathcal{T}_1 \cup \mathcal{T}_2$. We fix such a pair Γ-tree $\mathfrak{T}$. If $\mathfrak{T}$ is a singleton tree, then there exists a pair (t, U) with $U = \emptyset$ which is realizable in $\mathfrak{T}$. Hence, the algorithm accepts the input in Step 1. If no such singleton tree exists, then $\mathfrak{T}$ is a non-singleton tree and $\mathcal{R}_0 \neq \emptyset$. Let n be the depth of $\mathfrak{T}$. By definition, the sets $\mathcal{R}_i$, $i < n$, generated by the algorithm contain all internal pairs realizable in subtrees of $\mathfrak{T}$ generated by nodes of depth $\leq i$. Hence, a root pair (t, U) with $U = \emptyset$ is obtained in one step from $\mathcal{R}_{n-1}$. Clearly, for the first j with $\mathcal{R}_j = \mathcal{R}_j \cup \mathcal{R}'_j$, $\mathcal{R}_j$ contains $\mathcal{R}_{n-1}$ and, as a consequence, the algorithm accepts the input.

- Complexity: First observe that the number of internal and root pairs is bounded by
$$k_{\mathcal{T}_1,\mathcal{T}_2} = 2^{4\times|\mathcal{T}_1\cup\mathcal{T}_2|} \times 2^{2^{|\mathcal{T}_1\cup\mathcal{T}_2|}}.$$
Also, the number of subsets of a set S with $|S| \leq k_{\mathcal{T}_1,\mathcal{T}_2}$ of cardinality bounded by $m_{\mathcal{T}_1,\mathcal{T}_2}$ is at most doubly exponential in $|\mathcal{T}_1 \cup \mathcal{T}_2|$. Now,

using the fact that $\mathcal{ALCQI}$ satisfiability relative to TBoxes is decidable in ExpTime [11] and that realizability of root and internal pairs in a singleton Γ-tree can be checked using at most exponentially many satisfiability tests for concepts of polynomial size relative to $\mathcal{ALCQI}$-TBoxes, it follows that the time complexity of Step 1 and 2 of the algorithm is at most 2-exponential in $|\mathcal{T}_1 \cup \mathcal{T}_2|$. For Step 3., we obviously have $\mathcal{R}_i = \mathcal{R}_i \cup \mathcal{R}'_i$ for some $i \leq k_{\mathcal{T}_1,\mathcal{T}_2}$. Thus, the number of rounds of computations of $\mathcal{R}'_i$ from $\mathcal{R}_i$ is bounded by $k_{\mathcal{T}_1,\mathcal{T}_2}$. By Lemma 16, the computation of $\mathcal{R}'_i$ from $\mathcal{R}_i$ requires at most double exponentially many computations each of which requires only double exponential time. Also, by Lemma 16 checking whether a root pair (t, U) with $U = \emptyset$ is obtained in one step from a subset of $\mathcal{R}_i$ of cardinality bounded by $m_{\mathcal{T}_1,\mathcal{T}_2}$ requires not more than doubly exponential time. Hence, Step 3 does not require more than doubly exponential time.

❏

4 Undecidability in $\mathcal{ALCQIO}$

We show that conservative extensions are undecidable in $\mathcal{ALCQIO}$. The proof is by a reduction of the following undecidable tiling problem.

Definition 18 *A* domino system $D = (T, H, V, R, L, T, B)$ *consists of a finite set* T *of* tiles, *horizontal and vertical* matching relations $H, V \subseteq T \times T$, *and sets* $R, L, T, B \subseteq T$ *of* right *tiles,* left *tiles,* top *tiles, and* bottom *tiles. A* solution *to* D *is a pair* (n, m, τ) *where* $n, m \in \mathbb{N}$ *and* $\tau : n \times m \to T$ *such that the following hold:*

1. $(\tau(i,j), \tau(i+1,j)) \in H$, *for all* $i < n$ *and* $j \leq m$*;*
2. $(\tau(i,j), \tau(i,j+1)) \in V$, *for all* $i \leq n$ *and* $j < m$*;*
3. $\tau(0,j) \in L$ *and* $\tau(n,j) \in R$, *for all* $j \leq m$*;*
4. $\tau(i,0) \in B$ *and* $\tau(i,m) \in T$, *for all* $i \leq n$*.*

Using proof methods from [12], is is easy to show that it is undecidable whether a given domino system D has a solution. We show how to convert a domino system D into TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ such that D has a solution iff $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$. In particular, models of witness concepts will correspond to solutions of D.

Let $D = (T, H, V, R, L, T, B)$ be a domino system. The TBox $\mathcal{T}_1$ uses the following signature: an indidual name o, role names r_x and r_y, concept names top, bottom, left, and right and each element of T as a concept name. The TBox $\mathcal{T}_1$ contains the following:

- The roles r_x, r_y, and their inverses are functional:

$$\top \sqsubseteq (\leqslant 1\, r\, \top), \text{ for } r \in \{r_x, r_y, r_x^-, r_y^-\}$$

- Every position in the $n \times m$ grid is labeled with exactly one tile and the matching conditions are satisfied:

$$\begin{array}{lcl} \top & \sqsubseteq & \bigsqcup_{t \in T}(t \sqcap \bigsqcap_{t' \in T,\, t' \neq t} \neg t') \\ \top & \sqsubseteq & \bigsqcap_{t \in T}(t \rightarrow (\bigsqcup_{(t,t') \in H} \forall r_x.t' \sqcap \bigsqcup_{(t,t') \in V} \forall r_y.t')) \end{array}$$

- The concepts left, right, etc. mark the boundaries of the grid in the expected way:

$$\begin{array}{rcl} \mathsf{right} & \sqsubseteq & \neg \exists r_x.\top \sqcap \forall r_y.\mathsf{right} \sqcap \forall r_y^{-1}.\mathsf{right} \\ \neg\mathsf{right} & \sqsubseteq & \exists r_x.\top \end{array}$$

  and similarly for left, top, and bottom.

- The individual name o marks the origin:

$$\{o\} \sqsubseteq \mathsf{left} \sqcap \mathsf{bottom}.$$

The TBox $\mathcal{T}_2$ introduces two new concept names Q and P. It contains the following two concept inclusions:

$$\{o\} \sqsubseteq Q \sqsubseteq \exists r_x.Q \sqcup \exists r_y.Q \sqcup (\exists r_x.\exists r_y.P \sqcap \exists r_y.\exists r_x.\neg P)$$

The idea behind this definition of $\mathcal{T}_2$ is to enforce that models $\mathcal{I}$ of witness concepts are such that (i) there is no infinite outgoing r_x/r_y-path starting at $o^\mathcal{I}$ and (ii) r_x and r_y commute in the connected part of $\mathcal{I}$ rooted at $o^\mathcal{I}$. This is achieved as follows: if (i) is violated, then we can find an assignment of Q in $\mathcal{I}$ that satisfies $\mathcal{T}_2$. Similarly, if (ii) is violated, then we can find an assignment of (Q and) P in $\mathcal{I}$ that satisfies $\mathcal{T}_2$.

It can be checked that, as intended, D has a solution iff $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$. Here, we only show how to construct a witness concept in the case that D has a solution. Such a witness concept C has to ensure that for all models $\mathcal{I}$ of C and $\mathcal{T}_1$, the connected part of $\mathcal{I}$ rooted at $o^\mathcal{I}$ is isomorphic to the $n \times m$-grid.

For every word $w \in \{r_x, r_y\}^*$, denote by $\overleftarrow{w}$ the word that is obtained by reversing w and then adding $\cdot^-$ to each symbol. Let $|w|_r$ denote the number of occurrences of the symbol r in w. Now, C is the conjunction of

$$\{o\} \sqcap \forall r_x^n.\mathsf{right} \sqcap \forall r_y^m.\mathsf{top}$$

and for every $w \in \{r_x, r_y\}^*$ such that $|w|_{r_x} < n$ and $|w|_{r_y} < m$, the concept

$$\exists (w \cdot r_x r_y r_x^- r_y^- \cdot \overleftarrow{w}).\{o\},$$

where $\exists w.D$ abbreviates $\exists r_1. \cdots \exists r_k.D$ if $w = r_1 \cdots r_k$. It is not difficult to see that C enforces an $n \times m$-grid as required.

Theorem 19 *In $\mathcal{ALCQIO}$, conservative extensions are undecidable.*

Note that the theorem applies even to the case where $\Gamma = \mathsf{sig}(\mathcal{T}_1)$ and we allow $(\leqslant 1\ r\ \top)$ as the only form of number restriction.

5 Witness concepts

In this section, we provide a proof of the remaining implication of Lemma 7 and of the upper bound for the witness concept part of Theorem 4. For both proofs we associate with certain Γ-trees $\mathfrak{T}$ a concept (called diagram) C in $\mathcal{ALCQI}(\Gamma)$ which is satisfied in a tree model $\mathcal{I}$ if, and only if, there is a Γ-embedding of $\mathfrak{T}$ into $\mathcal{I}$. Such a diagram can be defined in a simple way only for trees in which the open nodes (elements of O) all have the same distance from the root, and this distance coincides with the depth of the tree. We require some notation.

Definition 20 (Level, balanced) *Let $\mathfrak{T} = (W, <, L, O)$ be a Γ-tree. The* level *$\ell(w)$ of a node w in $\mathfrak{T}$ is ∞ if $O = \emptyset$; otherwise it is n, where n is the length of the minimal path from w to an open node.*

A Γ-tree with root w whose depth does not exceed the level of w is called balanced. *In a balanced tree all open leaves have equal distance (namely the depth of the tree) to the root.*

Observe that the Γ-tree constructed in the proof of Lemma 7 is balanced. We are in a position now to define the concepts, called diagrams, describing Γ-trees.

Definition 21 (Diagram) *Let $\mathfrak{T} = (W, <, L, O)$ be a balanced Γ-tree with root w. Let m be the depth of $\mathfrak{T}$. We define, for each node v of level $k \leq m$ the* diagram of level n *($k \geq n \geq 0$) $\mathsf{diag}_n(\mathfrak{T}, v)$ of the pair $(\mathfrak{T}, v)$. The concept $\mathsf{diag}_n(\mathfrak{T}, v)$ is in $\mathcal{ALCQI}(\Gamma)$ and defined inductively as follows:*

- *set, for all nodes v, $\mathsf{diag}_0(\mathfrak{T}, v) = L(v)$;*
- *suppose $\mathsf{diag}_n(\mathfrak{T}, v)$ has been defined for all nodes v of level $\leq n$. Let u be a node of level $\geq n+1$. Then all its neighbors have level $\geq n$ and the n-diagram has been defined. Let, for every Γ-role r, $\mathcal{B}^r = \{A_1^r, \ldots, A_\ell^r\}$ be the partition of the set of r-neighbors of u such that $\mathsf{diag}_n(\mathfrak{T}, u') = \mathsf{diag}_n(\mathfrak{T}, u'')$ iff there exists $i \leq \ell$ with $u', u'' \in A_i^r$. Set*

$\mathsf{diag}_n(\mathfrak{T}, A_i^r) = \mathsf{diag}_n(\mathfrak{T}, u')$, for some $u' \in A_i^r$. Then $\mathsf{diag}_{n+1}(\mathfrak{T}, u)$ is defined as the conjunction of

$$\mathsf{diag}_n(\mathfrak{T}, u) \sqcap \bigsqcap_{r \text{ a } \Gamma\text{-role}} (= n_r\ r\ \top),$$

where n_r is the number of r-neighbors of u in $\mathfrak{T}$, and

$$\bigsqcap_{r \text{ a } \Gamma\text{-role}, A^r \in \mathcal{B}^r, m=|A^r|} (= m\ r\ \mathsf{diag}_n(\mathfrak{T}, A^r)).$$

The diagram $\mathsf{diag}(\mathfrak{T})$ is defined as $\mathsf{diag}_m(\mathfrak{T}, w)$.

Observe that a diagram of a balanced Γ-tree $\mathfrak{T}$ describes the $\mathfrak{T}$ in the following sense:

Lemma 22 *Let $\mathfrak{T} = (W, <, L, O)$ be a balanced Γ-tree with root w. Then the following conditions are equivalent, for all tree-models $\mathcal{I}$ with root $u \in \Delta^{\mathcal{I}}$:*

- *$u \in \mathsf{diag}(\mathfrak{T})^{\mathcal{I}}$;*
- *there is a Γ-embedding $f : \mathfrak{T} \to \mathcal{I}$.*

Proof. We start the proof with the following claim:

Claim 1. Let w and w' be two nodes in $\mathfrak{T}$ of level ℓ_w and $\ell_{w'}$, respectively. Take $\ell = \min(\ell_w, \ell_{w'})$. Then, for all $n \le \ell$, $\mathsf{diag}_n(\mathfrak{T}, w) \neq \mathsf{diag}_n(\mathfrak{T}, w')$ implies that $\mathsf{diag}_n(\mathfrak{T}, w) \sqcap \mathsf{diag}_n(\mathfrak{T}, w')$ is not satisfiable.

The proof is by a straighforward induction and left to the reader. Claim 1 is crucial for the inductive definition of the diagrams: it ensures that there are no "hidden" implications between syntactically different diagrams of nodes in a Γ-tree. In particular, there are no nodes to which diagrams which are syntactically different from its diagram apply. This ensures that the following can be proved by induction:

Claim 2. Let $f : \mathfrak{T} \to \mathcal{I}$ be a Γ-embedding. Then $u \in \mathsf{diag}(\mathfrak{T})^{\mathcal{I}}$ for the root u of $\mathcal{I}$.

Claim 2 implies one implication of Lemma 22. Conversely, the injection f required to show that for every tree model $\mathcal{I}$ with root u, $u \in \mathsf{diag}(\mathfrak{T})^{\mathcal{I}}$ implies that there is an Γ-embedding $f : \mathfrak{T} \to \mathcal{I}$ is easily constructed inductively from the root. ❑

We are now in the position to prove the remaining implication of Lemma 7: Suppose $\mathfrak{T} = (W, <, L, O)$ is a Γ-tree which is Γ-embeddable into a model of $\mathcal{T}_1$ but not into any model of $\mathcal{T}_1 \cup \mathcal{T}_2$. Let $f : \mathfrak{T} \to \mathcal{I}$ be a Γ-embedding into a tree model $\mathcal{I}$ of $\mathcal{T}_1$. We may assume that $\mathcal{I}$ has finite outdegree. If $\mathfrak{T}$ is balanced, then $\mathsf{diag}(\mathfrak{T})$ is a witness concept, by Lemma 22. Thus, $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a Γ-conservative extension of $\mathcal{T}_1$. If $\mathfrak{T}$ is not balanced, then let n be the depth of $\mathfrak{T}$. We hook to each $u \in O$ of distance $k < n$ from the root w

the labeled tree of depth at most $n - k$ induced by the set of points in $\mathcal{I}$ reachable from $f(u)$ in $n - k$-steps. (This can be done the same way as in the proof of Lemma 7.) The resulting Γ-tree is balanced and its diagram is a witness concept.

We are also in the position to prove the witness concept part of Theorem 4. Suppose $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a Γ-conservative extension of $\mathcal{T}_1$. Then the algorithm in Figure 2 above (implicity) constructs a Γ-tree $\mathcal{T} = (W, <, L, O)$ which is Γ-embeddable into a model $\mathcal{I}$ of $\mathcal{T}_1$ but not into any model for $\mathcal{T}_1 \cup \mathcal{T}_2$. The depth of this tree is a most double exponential, say m, in $|\mathcal{T}_1 \cup \mathcal{T}_2|$. Its outdegree is at most exponential in $|\mathcal{T}_1 \cup \mathcal{T}_2|$.

Of course, following the accepting computation, we can associate with each node v of $\mathfrak{T}$ a $\mathcal{T}_1$-type $t(v)$ such that there is a Γ-embedding $f : \mathfrak{T} \to \mathcal{I}$ into a tree model $\mathcal{I}$ of $\mathcal{T}_1$ and $t_{\mathcal{I}}^{\mathcal{T}_1}(f(v)) = t(v)$ holds for all v in $\mathfrak{T}$. We are now going to construct a balanced Γ-tree $\mathfrak{T}'$ from $\mathfrak{T}$ by hooking certain Γ-trees to leafs $v \in O$. Let v be a leaf in $\mathfrak{T}$ with $v \in O$. Let $k \leq m$ be the distance between v and the root w of $\mathfrak{T}$. If $k = m$, then we do nothing with v. Suppose $k < m$. Then k is not the root of $\mathfrak{T}$. There exists a Γ-tree $\mathfrak{T}_v = (W_v, <_v, L_v, O_v)$ and a model $\mathcal{I}^v$ of $\mathcal{T}_1$ such that, for the predecessor v' of v with $L(v', v) = r$ (for a Γ-role r) there are d', d in $\mathcal{I}^v$ with $d' \in t(v')^{\mathcal{I}^v}$, $d \in t(v)^{\mathcal{I}}$, $d' r^{\mathcal{I}^v} d$ and

- $\mathfrak{T}_v$ is Γ-embeddable into $\mathcal{I}_d^v$;
- the outdegree of $\mathfrak{T}_v$ is only exponential $|\mathcal{T}_1|$;
- either $\mathfrak{T}^v$ has depth $< m - k$ and $O^v = \emptyset$ or it has depth $= m - k$ and O^v is a subset of the set of leafs with distance $m - k$ from v.

Moreover, given t, t' and r such a Γ-tree can be constructed in triple exponential time in $|\mathcal{T}_1|$. Now hook to each such $v \in O$ such a Γ-tree $\mathfrak{T}^v$ and denote the resulting Γ-tree by $\mathfrak{T}'$. $\mathfrak{T}'$ is balanced and is embeddable into a model of $\mathcal{T}_1$ but not into any model of $\mathcal{T}_1 \cup \mathcal{T}_2$. The diagram of $\mathfrak{T}'$ is a witness concept of size at most three exponential in $|\mathcal{T}_1 \cup \mathcal{T}_2|$.

6 Conclusion

We have shown that conservative extensions are decidable and 2-ExpTime complete in $\mathcal{ALCQI}$, but undecidable in $\mathcal{ALCQIO}$. Although the high computational complexity suggests that efficient tools for deciding conservative extensions will be difficult to attain, our results and techniques lay theoretical foundations that are important for practical applications of conservative extensions. These could be based on approximations, semi-decision procedures, and on syntactic restrictions in a normative framework such as [3]. Finally, transitive roles are a main ingredient of DLs underlying OWL-DL.

It remains an important open problem to investigate conservative extensions for DLs containing transitive roles and role hierarchies.

References

[1] G. Antoniou and K. Kehagias. A note on the refinement of ontologies. *International Journal of Intelligent Systems*, 15:623–632, 2000.

[2] F. Baader, I. Horrocks, and U. Sattler. Description logics as ontology languages for the semantic web. In *Mechanizing Mathematical Reasoning*, volume 2605 of *LNAI*, pages 228–248. Springer, 2005.

[3] B. Cuenca-Grau, I. Horrocks, O. Kutz, and U. Sattler. Will my ontologies fit together? In *Proc. of DL06*, number 189 in CEUR-WS (http://ceur-ws.org/), 2006.

[4] S. Ghilardi, C. Lutz, and F. Wolter. Did I damage my ontology? A case for conservative extensions in description logic. In *Proc. of KR2006*, pages 187–197, 2006.

[5] B. Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularity and web ontologies. In *Proc. of KR2006*, pages 198–208, 2006.

[6] N. Guarino and C.A. Welty. An overview of OntoClean. In *Handbook of Ontologies*. Springer, 2004.

[7] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proc. of LPAR'99*, number 1705 in LNAI, pages 161–180. Springer, 1999.

[8] T. Maibaum. Conservative extensions, interpretations between theories and all that! In *Proc. of 7th CAAP/FASE Conf. on Theory and Practice of Software Development*, pages 40–66, 1997.

[9] S.K. Thomason. Reduction of tense logic to modal logic I. *Journal of Symbolic Logic*, 39:549–551, 1974.

[10] S.K. Thomason. The logical consequence relation of propositional tense logic. *Zeitschrift fuer Mathematische Logik und Grundlagen der Mathematik*, 21:29–40, 1975.

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

[12] P. van Emde Boas. The convenience of tilings. In A. Sorbi, editor, *Complexity, Logic, Recursion Theory*. 1997.