Conservative Extensions in the Lightweight Description Logic $\mathcal{EL}$

Carsten Lutz[1] and Frank Wolter[2]

[1] Institute for Theoretical Computer Science, TU Dresden, Germany
[2] Department of Computer Science, University of Liverpool, UK
lutz@tcs.inf.tu-dresden.de,frank@csc.liv.ac.uk

Abstract. We bring together two recent trends in description logic (DL): *lightweight DLs* in which the subsumption problem is tractable and *conservative extensions* as a central tool for formalizing notions of ontology design such as refinement and modularity. Our aim is to investigate conservative extensions as an automated reasoning problem for the basic tractable DL $\mathcal{EL}$. The main result is that deciding (deductive) conservative extensions is ExpTime-complete, thus more difficult than subsumption in $\mathcal{EL}$, but not more difficult than subsumption in expressive DLs. We also show that if conservative extensions are defined model-theoretically, the associated decision problem for $\mathcal{EL}$ is undecidable.

1 Introduction

In recent years, lightweight description logics (DLs) have gained increased popularity. There are two reasons for this. First, a number of useful lightweight DLs have been identified for which reasoning is tractable even w.r.t. general TBoxes (sets of subsumptions between concepts). Such TBoxes play a central role in almost all modern applications of DLs. And second, an increasing number of large-scale ontologies has been constructed for use in practical applications. Such ontologies usually require a high level of abstraction, and are often formulated in lightweight DLs.

Regarding the first point, there are mainly two lines of research: the $\mathcal{EL}$ family of tractable DLs investigated in [5, 2] aims at providing a logical underpinning of lightweight ontology languages, with a special emphasis on life science ontologies. In contrast, the main purpose of the DL-Lite family of tractable DLs investigated in [6, 7] is to allow efficient reasoning about conceptual database schemas, and to exploit existing DBMSs for DL reasoning. In this paper, we will be interested in applications of DLs for ontology design, and thus consider $\mathcal{EL}$ as our basic tractable DL. The main reasoning problem in $\mathcal{EL}$ is *subsumption*, i.e., deciding whether one concept subsumes another one w.r.t. a general TBox. Intuitively, such a TBox can be thought of as a logical theory providing a description of the application domain. In the following, we use the terms "general TBox" and "ontology" interchangeably.

There are a number of important life science ontologies that are formulated in $\mathcal{EL}$ or mild extensions thereof. Examples include the Systematized Nomenclature of Medicine, Clinical Terms (SNOMED CT), which comprises ∼0.5 million concepts and underlies the systematized medical terminology used in the health systems of the US, the UK, and other countries [15]; and the thesaurus of the US national cancer institute (NCI), which comprises ∼45.000 concepts and is intended to become the reference terminology for cancer research [14]. With ontologies of this size, a principled approach to their design and maintenance is indispensible, and automated reasoning support is highly welcome.

Recently, conservative extensions have been identified as a fundamental notion when formalizing central issues of ontology design such as refinement and modularity [1, 9, 12, 10, 11]. Unless otherwise noted, we refer to the deductive version of conservative extensions: the extension $\mathcal{T}_1 \cup \mathcal{T}_2$ of an ontology $\mathcal{T}_1$ is conservative if $\mathcal{T}_1 \cup \mathcal{T}_2$ implies no new subsumptions in the signature of $\mathcal{T}_1$, i.e., every subsumption $C \sqsubseteq D$ that is implied by $\mathcal{T}_1 \cup \mathcal{T}_2$ and where the concepts C and D use only symbols (concept and role names) from $\mathcal{T}_1$ is already implied by $\mathcal{T}_1$.

We briefly sketch how conservative extensions can help to formalize ontology refinement and modularity. *Refinement* means to add more details to a part of the ontology that has not yet been sufficiently described. Intuitively, such a refinement should provide more detailed information about the meaning of concepts of the original ontology, but it should not affect the relationship between such concepts. This requirement can be formalized by demanding that the refined ontology is a conservative extension of the original ontology. The main benefits of *modularity* of ontologies are that changes to the ontology have only local impact, and that modules from an ontology can be re-used in other ontologies. Intuitively, a module inside an ontology should be self-contained in the sense that it contains all the relevant information about the concepts it uses. Formally, this can be captured by requiring that a module inside an ontology $\mathcal{T}$ is a subset $\mathcal{T}'$ of $\mathcal{T}$ such that $\mathcal{T}$ is a conservative extension of $\mathcal{T}'$. See e.g.[10] for more details.

In [9, 12], it was proposed to provide automated reasoning support for conservative extensions. For example, if an ontology designer intends to refine his ontology, he may use an automated reasoning tool capable of deciding conservative extensions to check whether his modifications really had no impact on relationsships between concepts in the original ontology. The complexity of deciding conservative extensions is usually rather high. For example, it is 2-EXPTIME complete in expressive DLs such as $\mathcal{ALC}$ and $\mathcal{ALCQI}$ and even undecidable in $\mathcal{ALCQIO}$ [9, 12]; recall that subsumption is decidable in EXPTIME and, respectively, NEXPTIME for those logics.

In this paper, we study conservative extensions in the basic tractable description logic $\mathcal{EL}$. This is motivated by the observation that large-scale ontologies are often formulated in such lightweight DLs, and large-scale ontologies is also where issues of refinement and modularity play the most important role. We provide an alternative characterization of conservative extension in $\mathcal{EL}$, and use this characterization to provide a decision procedure. It is interesting to note that decision procedures for deciding conservative extensions in more expressive DLs such as $\mathcal{ALC}$ can *not* be used for $\mathcal{EL}$, see Section 2 for an example illustrating this effect. We show that our algorithm runs in deterministic exponential time, and prove a matching lower bound. Thus, deciding conservative extension in $\mathcal{EL}$ is ExpTime-complete and not tractable like subsumption in $\mathcal{EL}$. However, it is also not more difficult than subsumption in expressive DLs such as $\mathcal{ALC}$ and $\mathcal{ALCQI}$, problems that are considered manageable in practice. We also consider a stronger, model theoretic notion of conservative extensions that is useful for query answering and prove that the associated decision problem for $\mathcal{EL}$ is undecidable.

In this version of the paper, many proof details are omitted for brevity. They can be found in the full version [13].

2 $\mathcal{EL}$ and Conservative Extensions

Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be countably infinite and disjoint sets of *concept names* and *role names*, respectively. *$\mathcal{EL}$-concepts* C are built according to the syntax rule $C ::= \top \mid A \mid C \sqcap D \mid \exists r.C$, where A ranges over $\mathsf{N_C}$, r ranges over $\mathsf{N_R}$, and C, D range over $\mathcal{EL}$-concepts. The semantics is defined by means of an *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where the interpretation *domain* $\Delta^\mathcal{I}$ is a non-empty set, and $\cdot^\mathcal{I}$ is a function mapping each concept name A to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$ and each role name $r^\mathcal{I}$ to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The function $\cdot^\mathcal{I}$ is inductively extended to arbitrary concepts by setting $\top^\mathcal{I} := \Delta^\mathcal{I}$, $(C \sqcap D)^\mathcal{I} := C^\mathcal{I} \cap D^\mathcal{I}$, and $(\exists r.C)^\mathcal{I} := \{d \in \Delta^\mathcal{I} \mid \exists e \in C^\mathcal{I} : (d, e) \in r^\mathcal{I}\}$.

A *TBox* is a finite set of *concept inclusions (CIs)* $C \sqsubseteq D$, where C and D are concepts. An interpretation $\mathcal{I}$ *satisfies* a CI $C \sqsubseteq D$ (written $\mathcal{I} \models C \sqsubseteq D$) if $C^\mathcal{I} \subseteq D^\mathcal{I}$. $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ if it satisfies all CIs in $\mathcal{T}$. We write $\mathcal{T} \models C \sqsubseteq D$ if every model of $\mathcal{T}$ satisfies $C \sqsubseteq D$. Here is an example TBox $\mathcal{T}_1$:

$$\begin{aligned} \mathsf{Human} &\sqsubseteq \exists \mathsf{eats}.\top \\ \mathsf{Plant} &\sqsubseteq \exists \mathsf{grows\text{-}in}.\mathsf{Area} \\ \mathsf{Vegetarian} &\sqsubseteq \mathsf{Healthy} \end{aligned}$$

A *signature* Σ is a finite subset of $\mathsf{N_C} \cup \mathsf{N_R}$. The signature $\mathsf{sig}(C)$ ($\mathsf{sig}(\mathcal{T})$) of a concept C (TBox $\mathcal{T}$) is the set of concept and role names which occur in C (in $\mathcal{T}$). If $\mathsf{sig}(C) = \Sigma$, we also call C a *Σ-concept.* Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be TBoxes. We call $\mathcal{T}_1 \cup \mathcal{T}_2$ a *conservative extension* of $\mathcal{T}_1$ if $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq D$ implies $\mathcal{T}_1 \models C \sqsubseteq D$ for all $\mathsf{sig}(\mathcal{T}_1)$-concepts C, D. If C, D violate this condition (and thus, $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$), then $C \sqsubseteq D$ is called a *counter-subsumption.* As an example, consider the following TBox $\mathcal{T}_2$:

$$\begin{aligned} \mathsf{Human} &\sqsubseteq \exists \mathsf{eats}.\mathsf{Food} \\ \mathsf{Food} \sqcap \mathsf{Plant} &\sqsubseteq \mathsf{Vegetarian} \end{aligned}$$

It is not too difficult to verify that $\mathcal{T}_1 \cup \mathcal{T}_2$ is a conservative extension of $\mathcal{T}_1$, where $\mathcal{T}_1$ is the TBox defined above. Interestingly, the notion of a conservative extension strongly depends on the description logic used. For example, $\mathcal{ALC}$ is the extension of $\mathcal{EL}$ with a negation constructor $\neg C$, which has the obvious semantics $(\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I}$. In $\mathcal{ALC}$, $\forall r.C$ is an abbreviation for $\neg\exists r.\neg C$. If we view the TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ from above as $\mathcal{ALC}$ TBoxes, then $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$, with counter-subsumption

$$\mathsf{Human} \sqcap \forall \mathsf{eats}.\mathsf{Plant} \sqsubseteq \exists \mathsf{eats}.\mathsf{Vegetarian}.$$

This shows that we cannot use the existing algorithms for conservative extensions in $\mathcal{ALC}$ [9] to decide conservative extension in $\mathcal{EL}$.

Another initial observation about conservative extensions in $\mathcal{EL}$ is that minimal counter-subsumptions may be quite large. Define a TBox $\mathcal{T}$ such that it contains only tautologies and $\mathsf{sig}(\mathcal{T}_1) = \{A, B, r, s\}$. For each $n \geq 0$, we define a TBox $\mathcal{T}'_n$. It has additional concept names $X_0, \ldots, X_{n-1}$ and $\overline{X}_0, \ldots, \overline{X}_{n-1}$ that are used to represent a binary counter X: if X_i is true, then the i-th bit is positive and if $\overline{X}_i$ is true, then it is negative. Define $\mathcal{T}'_n$ as

$$\begin{array}{rll}
A \sqsubseteq \overline{X}_0 \sqcap \cdots \sqcap \overline{X}_{n-1} & & \\
\bigsqcap_{\sigma \in \{r,s\}} \exists\sigma.(\overline{X}_i \sqcap X_0 \sqcap \cdots \sqcap X_{i-1}) \sqsubseteq X_i & & \text{for all } i < n \\
\bigsqcap_{\sigma \in \{r,s\}} \exists\sigma.(X_i \sqcap X_0 \sqcap \cdots \sqcap X_{i-1}) \sqsubseteq \overline{X}_i & & \text{for all } i < n \\
\bigsqcap_{\sigma \in \{r,s\}} \exists\sigma.(\overline{X}_i \sqcap \overline{X}_j) \sqsubseteq \overline{X}_i & & \text{for all } j < i < n \\
\bigsqcap_{\sigma \in \{r,s\}} \exists\sigma.(X_i \sqcap \overline{X}_j) \sqsubseteq X_i & & \text{for all } j < i < n \\
X_0 \sqcap \cdots \sqcap X_{n-1} \sqsubseteq B & &
\end{array}$$

Observe that Lines 2-5 implement incrementation of the counter X. Then the smallest new consequence of $\mathcal{T} \cup \mathcal{T}'_n$ is $C_{2^n} \sqsubseteq B$, where:

$$\begin{aligned}
C_0 &= A \\
C_i &= \exists r.C_{i-1} \sqcap \exists s.C_{i-1}
\end{aligned}$$

Clearly, C_{2^n} is doubly exponentially large in the size of $\mathcal{T}$ and $\mathcal{T}'_n$. If we use structure sharing (i.e., define the size of C_{2^n} as the number of its distinct subconcepts), it is still exponentially large.

3 Characterizing Conservative Extensions

In this section, we provide a characterization of when a TBox $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$. In the subsequent section, we will use this characterization to devise a decision procedure for (non-)conservative extensions in $\mathcal{EL}$.

Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be interpretations and Σ a signature. A relation $S \subseteq \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$ is a *Σ-simulation* from $\mathcal{I}_1$ to $\mathcal{I}_2$ if the following holds:

- for all concept names $A \in \Sigma$ and all $(d_1, d_2) \in S$ with $d_1 \in A^{\mathcal{I}_1}$, $d_2 \in A^{\mathcal{I}_2}$;
- for all role names $r \in \Sigma$, all $(d_1, d_2) \in S$, and all $e_1 \in \Delta^{\mathcal{I}_1}$ with $(d_1, e_1) \in r^{\mathcal{I}_1}$, there exists $e_2 \in \Delta^{\mathcal{I}_2}$ such that $(d_2, e_2) \in r^{\mathcal{I}_2}$ and $(e_1, e_2) \in S$.

If $d_1 \in \Delta^{\mathcal{I}_1}$, $d_2 \in \Delta^{\mathcal{I}_2}$, and there is a Σ-simulation S with $(d_1, d_2) \in S$, then $(\mathcal{I}_2, d_2)$ *Σ-simulates* $(\mathcal{I}_1, d_1)$, written $(\mathcal{I}_1, d_1) \leq_\Sigma (\mathcal{I}_2, d_2)$. If $\Sigma = \mathsf{N_C} \cup \mathsf{N_R}$, we simply speak of a simulation and write $\leq$ instead of $\leq_\Sigma$. Let $\mathcal{I}$ be an interpretation, Σ a signature, and $d \in \Delta^{\mathcal{I}}$. Then we define the abbreviation $d^{\Sigma,\mathcal{I}} := \{C \mid d \in C^{\mathcal{I}} \wedge \mathsf{sig}(C) \subseteq \Sigma\}$. The following theorem establishes a connection between simulations and $\mathcal{EL}$ formulas. The proof is standard, and therefore omitted, see e.g. [8].

Theorem 1. *If $(\mathcal{I}_1, d_1) \leq_\Sigma (\mathcal{I}_2, d_2)$, then $d_1^{\Sigma,\mathcal{I}_1} \subseteq d_2^{\Sigma,\mathcal{I}_2}$. Conversely, if $\mathcal{I}_1$ and $\mathcal{I}_2$ have finite out-degree and $d_1^{\Sigma,\mathcal{I}_1} \subseteq d_2^{\Sigma,\mathcal{I}_2}$, then $(\mathcal{I}_1, d_1) \leq_\Sigma (\mathcal{I}_2, d_2)$.*

We use $\mathsf{sub}(C)$ and $\mathsf{sub}(\mathcal{T})$ to denote the set of subconcepts of a concept C and TBox $\mathcal{T}$, respectively. With each concept C and TBox $\mathcal{T}$, we associate two sets of consequences that will play a central role in what follows.

- $K_\mathcal{T}(C) = \{D \in \mathsf{sub}(\mathcal{T}) \mid \mathcal{T} \models C \sqsubseteq D\}$;
- $L_\mathcal{T}(C) = \{D \in \mathsf{sub}(C) \mid \mathcal{T} \models C \sqsubseteq D\} \cup K_\mathcal{T}(C)$.

By the results in [5], both sets can be computed in time polynomial in the size of C and $\mathcal{T}$. The *canonical model* $\mathcal{I}_{C,\mathcal{T}} = (\Delta^{C,\mathcal{T}}, \cdot^{C,\mathcal{T}})$ of C and $\mathcal{T}$ is defined as follows:

- $\Delta^{C,\mathcal{T}} = \{C\} \cup \{C' \mid \exists r.C' \in \mathsf{sub}(C) \cup \mathsf{sub}(\mathcal{T})\}$;
- $D \in A^{\mathcal{I}_{C,\mathcal{T}}}$ iff $A \in L_\mathcal{T}(D)$, for all $A \in \mathsf{N_C}$;
- $(D, D') \in r^{\mathcal{I}_{C,\mathcal{T}}}$ iff $\exists r.D' \in K_\mathcal{T}(D)$ or $\exists r.D'$ is a conjunct in D, for all $r \in \mathsf{N_R}$.

When we say "$\exists r.D'$ is a conjunct in D", this also includes the case $D = \exists r.D'$. The model $\mathcal{I}_{C,\mathcal{T}}$ is a subtle refinement of the data structure generated by the algorithms in [5, 2] to prove correctness of the algorithm in [2].[1] Since the sets $L_\mathcal{T}(C)$ and $K_\mathcal{T}(C)$ can be computed in polytime, the model $\mathcal{I}_{C,\mathcal{T}}$ can can also be computed in time polynomial in the size of C and $\mathcal{T}$.

Lemma 1. *Let $\mathcal{T}$ be a TBox and C a concept. For all $D \in \Delta^{C,\mathcal{T}}$ and all $E \in \mathsf{sub}(C) \cup \mathsf{sub}(\mathcal{T})$, we have $D \in E^{\mathcal{I}_{C,\mathcal{T}}}$ iff $\mathcal{T} \models D \sqsubseteq E$.*

Lemma 1 implies that $\mathcal{I}_{C,\mathcal{T}}$ is a model of $\mathcal{T}$, and that $C \in C^{\mathcal{I}_{C,\mathcal{T}}}$. The following lemma summarizes the most important properties of canonical models. Regarding Points 1 and 2, similar (but simpler) lemmas for the case of $\mathcal{EL}$ without TBoxes have been established in [3].

Lemma 2. *Let C, C_1, C_2, D be $\mathcal{EL}$-concepts and $\mathcal{T}$ a TBox. Then the following holds:*

1. *For all models $\mathcal{I}$ of $\mathcal{T}$ and all $d \in \Delta^{\mathcal{I}}$, the following conditions are equivalent:*
   (a) $d \in C^{\mathcal{I}}$;
   (b) $(\mathcal{I}_{C,\mathcal{T}}, C) \leq (\mathcal{I}, d)$.
2. *The following conditions are equivalent:*
   (a) $\mathcal{T} \models C \sqsubseteq D$;

[1] Essentially, in those papers we have $(D, D') \in r^{\mathcal{I}_{C,\mathcal{T}}}$ iff $\exists r.D' \in L_\mathcal{T}(D)$.

(b) $C \in D^{\mathcal{I}_{C,\mathcal{T}}}$;
(c) $(\mathcal{I}_{D,\mathcal{T}}, D) \leq (\mathcal{I}_{C,\mathcal{T}}, C)$.

3. *If* $\exists r.D \in (\mathsf{sub}(C_i) \cup \mathsf{sub}(\mathcal{T}))$ *for all* $i \in \{1,2\}$, *then* $(\mathcal{I}_{C_1,\mathcal{T}}, D) \leq (\mathcal{I}_{C_2,\mathcal{T}}, D)$.

Let $\mathcal{T}_1$, $\mathcal{T}_2$ be TBoxes, C a $\mathsf{sig}(\mathcal{T}_1)$-concept, and D a $\mathsf{sig}(\mathcal{T}_1) \cup \mathsf{sig}(\mathcal{T}_2)$-concept. We write $C \Rightarrow_1 D$ if, for all $\mathsf{sig}(\mathcal{T}_1)$-concepts E, $\mathcal{T}_1 \cup \mathcal{T}_2 \models D \sqsubseteq E$ implies $\mathcal{T}_1 \models C \sqsubseteq E$. Our characterization of non-conservative extensions, as stated by the following lemma, is based on this relation. The main benefit of this characterization is that when checking for new subsumptions $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq D$, it allows us to concentrate on concepts D of a very simply form, namely subconcepts of $\mathcal{T}_1$ and $\mathcal{T}_2$. This is achieved by considering $\mathsf{sig}(\mathcal{T}_1) \cup \mathsf{sig}(\mathcal{T}_2)$-concepts instead of $\mathsf{sig}(\mathcal{T}_1)$-concepts as in the definition of conservative extensions. In addition, the characterization provides a bound on the *outdegree* of C, i.e., the maximum cardinality of any set P of pairs of the form (r, C'), with r a role name and C' a concept, such that $\bigsqcap_{(r,C') \in P} \exists r.C' \in \mathsf{sub}(C)$. We use $|C|$ and $|\mathcal{T}|$ to denote the *length* of a C and a TBox $\mathcal{T}$, i.e., the number of symbols needed to write it.

Lemma 3. *$\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$ iff there exists a $\mathsf{sig}(\mathcal{T}_1)$-concept C and a $\mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$-concept $D \in \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2)$ such that*

(a) $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq D$;
(b) $C \not\Rightarrow_1 D$;
(c) the outdegree of C is bounded by $|\mathcal{T}_1 \cup \mathcal{T}_2|$.

Proof. "⇒". Assume that (a) to (c) are satisfied. By (b), there is a concept E with $\mathcal{T}_1 \cup \mathcal{T}_2 \models D \sqsubseteq E$ and $\mathcal{T}_1 \not\models C \sqsubseteq E$. From the former and (a), we get $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq E$, which implies that $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$.

"⇐". We give only a sketch and refer to the full version [13] for details. Assume that $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$. In this sketch, we show only (a) and (b). If there is a counter-subsumption $C \sqsubseteq D$ with $D \in \mathsf{sub}(\mathcal{T}_1)$, then conditions (a) and (b) hold for C and D and we are done. Assume that no such counter-subsumption exists. Let $C \sqsubseteq D$ be a counter-subsumption such that D is of minimal length. Then D can be shown to be of the form $\exists r.D'$. Using Lemma 2, it is possible to prove that $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq \exists r.D'$ implies that one of the following holds:[2]

1. there is a conjunct $\exists r.C'$ of C such that $\mathcal{T}_1 \cup \mathcal{T}_2 \models C' \sqsubseteq D'$;
2. there is $\exists r.C' \in \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2)$ s.t. $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq \exists r.C'$ and $\mathcal{T}_1 \cup \mathcal{T}_2 \models C' \sqsubseteq D'$.

It is possible to show that Case 1 actually yields a contradiction to the minimal length of D. Thus, Case 2 applies. We show that the concepts C and $\exists r.C'$ (substituted for D) satisfy Conditions (a) and (b). First, $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq \exists r.C'$ establishes Condition (a). For Condition (b), observe that $\mathcal{T}_1 \not\models C \sqsubseteq \exists r.D'$ and $\mathcal{T}_1 \cup \mathcal{T}_2 \models \exists r.C' \sqsubseteq \exists r.D'$. This means $C \not\Rightarrow_1 \exists r.C'$. ❏

[2] Without the refined version of canonical models, this would not be true.

The following lemma characterizes the relation $C \Rightarrow_1 D$ semantically and shows that it can be decided in polytime.

Lemma 4. *Let $\mathcal{T}_1, \mathcal{T}_2$ be TBoxes and C, D concepts. Then we have $C \Rightarrow_1 D$ iff $\mathcal{I}_{D,\mathcal{T}_1 \cup \mathcal{T}_2} \leq_{\mathsf{sig}(\mathcal{T}_1)} \mathcal{I}_{C,\mathcal{T}_1}$. Hence, the problem $C \Rightarrow_1 D$ is decidable in polynomial time in the size of C, D, and $\mathcal{T}_1 \cup \mathcal{T}_2$.*

Proof. "$\Rightarrow$". Let $C \not\Rightarrow_1 D$. Then there is a $\mathsf{sig}(\mathcal{T}_1)$-concept E such that $\mathcal{T}_1 \cup \mathcal{T}_2 \models D \sqsubseteq E$ and $\mathcal{T}_1 \not\models C \sqsubseteq E$. By Point 2 of Lemma 2, this yields $D \in E^{\mathcal{I}_{D,\mathcal{T}_1 \cup \mathcal{T}_2}}$ and $C \notin E^{\mathcal{I}_{C,\mathcal{T}_1}}$. Hence, by Theorem 1, $(\mathcal{I}_{D,\mathcal{T}_1 \cup \mathcal{T}_2}, D) \not\leq_{\mathsf{sig}(\mathcal{T}_1)} (\mathcal{I}_{C,\mathcal{T}_1}, C)$.

"$\Leftarrow$". Let $(\mathcal{I}_{D,\mathcal{T}_1 \cup \mathcal{T}_2}, D) \not\leq_{\mathsf{sig}(\mathcal{T}_1)} (\mathcal{I}_{C,\mathcal{T}_1}, C)$. By Theorem 1, there exists E over $\mathsf{sig}(\mathcal{T}_1)$ with $D \in E^{\mathcal{I}_{D,\mathcal{T}_1 \cup \mathcal{T}_2}}$ but $C \notin E^{\mathcal{I}_{C,\mathcal{T}_1}}$. By Point 2 of Lemma 2, $\mathcal{T}_1 \cup \mathcal{T}_2 \models D \sqsubseteq E$ and $\mathcal{T}_1 \not\models C \sqsubseteq E$. Hence, $C \not\Rightarrow_1 D$. It is well-known that computing the largest Σ-simulation between two finite graphs can be done in polynomial time [8]. ❑

4 The Algorithm

We devise an algorithm for deciding (non)-conservative extensions in $\mathcal{EL}$, which is based on our characterization of not being a conservative extensions in terms of "$\Rightarrow_1$" (Lemma 3) and of "$\Rightarrow_1$" in terms of simulations (Lemma 4). To check whether $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$, the algorithm searches for a $\mathsf{sig}(\mathcal{T}_1)$-concept C such that for some $D \in \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2)$, the Points (a)–(c) of Lemma 3 are satisfied. Intuitively, it proceeds in rounds. In the first round, the algorithm considers the case where C is a conjunction of concept names. For every such C and all $D \in \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2)$, it checks whether Points (a) and (b) are satisfied. By Lemma 4, this can be done in polytime. If all tests fail, the second round is started in which the algorithm considers concepts C of the form $F_0 \sqcap \bigsqcap_{(r,E) \in P} \exists r.E$, where F_0 is a conjunction of concept names and P is a set of pairs (r, E) with r a role name and E a candidate for C from the first round (i.e., E is also a conjunction of concept names). Because of Point (c), it will be sufficient to consider sets P of cardinality bounded by $|\mathcal{T}_1 \cup \mathcal{T}_2|$. To check if such a concept C satisfies Points (a) and (b), we exploit the information that we have gained about the concepts E in the previous round. If again no suitable C is found, then in the third round we use the Cs from the second round as the Es in $F_0 \sqcap \bigsqcap_{(r,E) \in P} \exists r.E$, and so on.

For the algorithm to terminate and run in exponential time, we have to introduce a condition that indicates when enough candidates C have been inspected in order to know that there is no counter-subsumption $C \sqsubseteq D$. To obtain such a termination condition and to avoid having to deal with double exponentially large concepts, our algorithm will not construct the candidate concepts C directly, but rather use a certain a data structure to represent relevant information about C. The relevant information about C is suggested by Lemma 3: for each C, we take the quadruple

$$C^\sharp = (F, K_{\mathcal{T}_1}(C), K_{\mathcal{T}_1 \cup \mathcal{T}_2}(C), K_{\mathcal{T}_1, \mathcal{T}_1 \cup \mathcal{T}_2}(C)),$$

Imput: TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$.

1. Compute the set $\mathcal{N}_0$ of quadruples determined by conjunctions of concept names from $\mathsf{sig}(\mathcal{T}_1)$.
2. if $\mathcal{N}_0$ contains a quadruple $(F, \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3)$ such that $\mathcal{Q}_2 \setminus \mathcal{Q}_3 \neq \emptyset$, then output "not conservative extension".
3. Generate the sequence $\mathcal{N}_1, \mathcal{N}_2, \ldots$ of quadruples such that $\mathcal{N}_{i+1} = \mathcal{N}_i \cup \mathcal{N}_i'$, where $\mathcal{N}_i'$ is the set of quadruples $(F_0, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3)$ which can be obtained from a conjunction F_0 of concept names from $\mathsf{sig}(\mathcal{T}_1)$ and a set $Q \subseteq (\mathsf{N_R} \cap \mathsf{sig}(\mathcal{T}_1)) \times \mathcal{N}_i$ of cardinality not exceeding $|\mathcal{T}_1 \cup \mathcal{T}_2|$ in the following way:
   - $\mathcal{F}_1 = K_{\mathcal{T}_1}(F_0 \sqcap \bigsqcap_{(r,(F,\mathcal{Q}_1,\mathcal{Q}_2,\mathcal{Q}_3)) \in Q} \exists r.(\bigsqcap_{D \in \mathcal{Q}_1} D))$;
   - $\mathcal{F}_2 = K_{\mathcal{T}_1 \cup \mathcal{T}_2}(F_0 \sqcap \bigsqcap_{(r,(F,\mathcal{Q}_1,\mathcal{Q}_2,\mathcal{Q}_3)) \in Q} \exists r.(\bigsqcap_{D \in \mathcal{Q}_2} D))$;
   - $\mathcal{F}_3 = \{D \mid D \in \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2)$ and
     (a) for all $A \in \mathsf{sig}(\mathcal{T}_1)$, $A \in K_{\mathcal{T}_1 \cup \mathcal{T}_2}(D)$ implies $A \in \mathcal{F}_1$;
     (b) if $(D, D') \in r^{\mathcal{I}_{D,\mathcal{T}_1 \cup \mathcal{T}_2}}$ with $r \in \mathsf{sig}(\mathcal{T}_1)$, then
     (i) there is a tuple $(r, (F, \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3)) \in Q$ such that $D' \in \mathcal{Q}_3$
     or (ii) there is $\exists r.C' \in \mathcal{F}_1$ with $(\mathcal{I}_{D',\mathcal{T}_1 \cup \mathcal{T}_2}, D') \leq_{\mathsf{sig}(\mathcal{T}_1)} (\mathcal{I}_{C',\mathcal{T}_1}, C')$
     $\}$

   This is done until $\mathcal{N}_i$ contains a quadruple $(F, \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3)$ such that $\mathcal{Q}_2 \setminus \mathcal{Q}_3 \neq \emptyset$, or $\mathcal{N}_{i+1} = \mathcal{N}_i$. Output "not conservative extension" if the first condition applies. Otherwise, output "conservative extension".

Fig. 1. Algorithm for deciding (non)-conservative extensions in $\mathcal{EL}$

where F is the conjunction of all concept names occurring in the top-level conjunction of C (if there are none, then $F = \top$), $K_{\mathcal{T}_1 \cup \mathcal{T}_2}(C)$ is defined in the previous section, and $K_{\mathcal{T}_1, \mathcal{T}_1 \cup \mathcal{T}_2}(C) = \{D \in \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2) \mid C \Rightarrow_1 D\}$. We call this the *quadruple determined by* C.

By Lemma 3, the quadruple $C^\sharp$ determined by a concept C gives us enough information to decide whether C is the left hand side of a counter-subsumption. In addition, it contains enough information to enable the recursive search described above. This is exploited by our algorithm for deciding (non)-conservative extensions, which is shown in Figure 1. Observe that the Condition $\mathcal{Q}_2 \setminus \mathcal{Q}_3 \neq \emptyset$ corresponds to satisfaction of Points (a) and (b) in Lemma 3. Also observe that, in Point (b) of the definition of $\mathcal{F}_3$, we refer to the canonical model $\mathcal{I}_{D,\mathcal{T}_1 \cup \mathcal{T}_2}$ for the relevant concepts D. These models are constructed in polytime when needed. To show that this algorithm really implements the initial description given at the beginning of this section, we make explicit the concepts that we describe by means of the quadruples constructed in Step 3 of Figure 1. This is done by the following lemma, which will also be a central ingredient to our correctness proof.

Lemma 5. *Let* $(F_0, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3)$ *be the quadruple obtained from* F_0 *and* Q *in Figure 1. Let, for each* $(r, q) \in Q$, $C_{r,q}$ *be a concept which determines* q. *Then* $C = F_0 \sqcap \bigsqcap_{(r,q) \in Q} \exists r.C_{r,q}$ *determines* $(F_0, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3)$.

Proof. Let $(F_0, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3)$ and C be as in the lemma. It is trivial to see that F_0 is as required. To treat $\mathcal{F}_1$ and $\mathcal{F}_2$, we prove the following in [13]: for all TBoxes $\mathcal{T}$ and concepts $C' = F_0' \sqcap \bigsqcap_{(r,E)\in P} \exists r.E$ with F_0' a conjunction of concept names,

$$K_\mathcal{T}(C') = K_\mathcal{T}(F_0' \sqcap \bigsqcap_{(r,E)\in P} \exists r.(\bigsqcap_{D\in K_\mathcal{T}(E)} D)).$$

This implies that $\mathcal{F}_1$ and $\mathcal{F}_2$ are as required. It remains to consider $\mathcal{F}_3$. Fix $D \in \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2)$. By Lemma 4, $C \Rightarrow_1 D$ iff $(\mathcal{I}_{D,\mathcal{T}_1\cup\mathcal{T}_2}, D) \leq_{\mathsf{sig}(\mathcal{T}_1)} (\mathcal{I}_{C,\mathcal{T}_1}, C)$. By definition of simulations and once more by Lemma 4, to check whether $C \Rightarrow_1 D$ it is sufficient to check both of the following:

1. for all concept names $A \in \mathsf{sig}(\mathcal{T}_1)$, $A \in K_{\mathcal{T}_1\cup\mathcal{T}_2}(D)$ implies $A \in K_{\mathcal{T}_1}(C)$;
2. for all $r \in \mathsf{sig}(\mathcal{T}_1)$ and D' with $(D, D') \in r^{\mathcal{I}_{D,\mathcal{T}_1\cup\mathcal{T}_2}}$ there exists C' with $(C, C') \in r^{\mathcal{I}_{C,\mathcal{T}_1}}$ and $(\mathcal{I}_{D,\mathcal{T}_1\cup\mathcal{T}_2}, C') \leq_{\mathsf{sig}(\mathcal{T}_1)} (\mathcal{I}_{C,\mathcal{T}_1}, D')$.

Point 1 is checked under (a) since, as we have seen already, $K_{\mathcal{T}_1}(C) = \mathcal{F}_1$. For Point 2, $(C, C') \in r^{\mathcal{I}_{C,\mathcal{T}_1}}$ and the definition of canonical models implies that we have (i) $\exists r.C'$ is a conjunct of C or (ii) $\exists r.C' \in K_{\mathcal{T}_1}(C)$. In Case (i), $C' = C_{r,q}$ for some $(r, q) \in Q$ and $C' \Rightarrow_1 D'$ iff D' is an element of the fourth component of q. This is what is checked in (b.i) of the algorithm. In Case (ii), $\exists r.C' \in \mathsf{sub}(\mathcal{T}_1)$ and thus we can use Point 3 of Lemma 1 to show that $(\mathcal{I}_{D,\mathcal{T}_1\cup\mathcal{T}_2}, C') \leq_{\mathsf{sig}(\mathcal{T}_1)} (\mathcal{I}_{C,\mathcal{T}_1}, D')$ iff $(\mathcal{I}_{D',\mathcal{T}_1\cup\mathcal{T}_2}, D') \leq_{\mathsf{sig}(\mathcal{T}_1)} (\mathcal{I}_{C',\mathcal{T}_1}, C')$. This is exactly what is checked in (b.ii) of the algorithm. ❑

Theorem 2. *The algorithm for deciding non-conservative extensions is sound, complete, and runs in exponential time.*

Proof. Soundness follows from Lemmas 3 and 5. For completeness, assume that $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$. By Lemma 3, there exists C of outdegree not exceeding $|\mathcal{T}_1 \cup \mathcal{T}_2|$ and $D \in \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2)$ such that $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq D$ and $C \not\Rightarrow_1 D$. If C is a conjunction of concept names, then the algorithm outputs "not conservative extension" in Step 2. Now suppose C has quantifier depth $n \geq 1$. Using Lemma 5, one can easily show by induction on i that for all $i \geq 0$, the set $\mathcal{N}_i$ contains all quadruples determined by subconcepts C' of C of quantifier depth smaller than i. Hence, the algorithm outputs "not conservative extension" after computing some $\mathcal{N}_i$ with $i \leq n$.

For termination and complexity, observe that, by Lemma 4, the quadruple determined by a conjunction of concept names from $\mathsf{sig}(\mathcal{T}_1)$ can be computed in polytime. Hence Steps 1 and 2 run in exponential time. For Step 3 observe that the number of tuples $(F, \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3)$ with F a conjunction of concept names from $\mathsf{sig}(\mathcal{T}_1)$ and $\mathcal{Q}_i \subseteq \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2)$ is bounded by $2^{4|\mathcal{T}_1\cup\mathcal{T}_2|}$. It follows that $\mathcal{N}_i = \mathcal{N}_{i+1}$ for some $i \leq 2^{4|\mathcal{T}_1\cup\mathcal{T}_2|}$. Hence, the algorithm terminates and to show that it runs in exponential time it remains to check that $\mathcal{N}_{i+1}$ can be computed in exponential time from $\mathcal{N}_i$. This follows from the following: first, the number of pairs (F_0, Q), with F_0 a conjunction of concept names from $\mathsf{sig}(\mathcal{T}_1)$ and $Q \subseteq (\mathsf{N_R} \cap \mathsf{sig}(\mathcal{T}_1)) \times \mathcal{N}_i$ of cardinality not exceeding $|\mathcal{T}_1 \cup \mathcal{T}_2|$, is still only exponential in $|\mathcal{T}_1 \cup \mathcal{T}_2|$; and second, the computation of $(F_0, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3)$ from F_0 and Q in Figure 1 can be done in time polynomial in $|\mathcal{T}_1 \cup \mathcal{T}_2|$. ❑

5 ExpTime-hardness

We prove ExpTime-hardness of deciding conservative extensions in $\mathcal{EL}$ by reduction of the problem of determining whether a given player has a winning strategy in the two-player game Peek introduced in [16] (the version G_4). An instance of Peek is a tuple $(\Gamma_1, \Gamma_2, \Gamma_I, \varphi)$ where:

- Γ_1 and Γ_2 are disjoint, finite sets of Boolean variables, with the intended interpretation that the variables in Γ_1 are under the control of Player 1, and Γ_2 is under the control of Player 2;
- $\Gamma_I \subseteq (\Gamma_1 \cup \Gamma_2)$ are the variables deemed to be true in the initial state of the game;
- φ is a propositional logic formula over the variables $\Gamma_1 \cup \Gamma_2$, representing the winning condition.

The game is played in a series of rounds, with the Players $i \in \{1, 2\}$ alternating (Player 1 moves first) to select a variable from Γ_i whose truth value is then flipped to reach the next game configuration. The game starts from the initial assignment defined by Γ_I. Variables that were not changed retain the same truth value in the subsequent configuration. A player may also make a skip move, i.e., not change any of its variables. He wins in a given round if he makes a move such that the resulting truth assignment defined by that round makes the winning formula φ true. The decision problem associated with Peek is to determine whether Player 2 has a winning strategy in a given game instance $(\Gamma_1, \Gamma_2, \Gamma_I, \varphi)$.

Given a game instance $G = (\Gamma_1, \Gamma_2, \Gamma_I, \varphi)$, we define TBoxes $\mathcal{T}_G$ and $\mathcal{T}'_G$ such that $\mathcal{T}_G \cup \mathcal{T}'_G$ is not a conservative extension of $\mathcal{T}_G$ iff Player 2 has a winning strategy in G. More precisely, $\mathcal{T}_G$ and $\mathcal{T}'_G$ are crafted such that witness subsumptions $C \sqsubseteq D$ against conservativity are such that (D is a concept name and) C describes a winning stategy for Player 2. Conversely, every winning strategy can be converted into a witness subsumption against conservativity. For convenience, we assume that the set of variables $\Gamma_1 \cup \Gamma_2$ is of the form $\{0, \dots, n-1\}$ for some $n \geq 2$. In $\mathcal{T}_G$, we use the following concept names to describe winning strategies.

- $V_0, \dots, V_{n-1}$ and $\overline{V}_0, \dots, \overline{V}_{n-1}$ to describe the truth values of the variables;
- $F_0, \dots, F_n$ to denote the variable that is flipped to reach the current configuration, with F_n indicating a skip move;
- P_1, P_2 to denote the player which moved to reach the current configuration.

We also use an additional concept name B, which plays a special role: we will construct $\mathcal{T}_G$ and $\mathcal{T}'_G$ such that if $\mathcal{T}_G \cup \mathcal{T}'_G$ is not a conservative extension of $\mathcal{T}_G$, then there is a witness subsumption $C \sqsubseteq D$ with $D = B$. Additionally, $\mathcal{T}_G$ uses a single role name r. We now assemble $\mathcal{T}_G$. We first say that the players alternate:

$$\exists r.P_1 \sqsubseteq P_2$$
$$\exists r.P_2 \sqsubseteq P_1$$

Then, we say that P_1 and P_2 should be disjoint. The idea is as follows: every concept C which enforces to make both P_1 and P_2 true somewhere in the model subsumes the special concept name B already w.r.t. $\mathcal{T}_G$, and thus cannot occur on the left-hand side of a witness subsumtion $C \sqsubseteq B$. The concept name M is used as a marker:

$$\begin{aligned} P_1 \sqcap P_2 &\sqsubseteq M \\ \exists r.M &\sqsubseteq M \\ M &\sqsubseteq B \end{aligned}$$

We also need disjointness conditions for truth values and flipping markers:

$$\begin{aligned} V_i \sqcap \overline{V}_i &\sqsubseteq M \quad \text{for all } i < n \\ F_i \sqcap F_j &\sqsubseteq M \quad \text{for all } i, j \leq n \text{ with } i \neq j \end{aligned}$$

Next, we say that if the marker F_i is set, the variable V_i flips:

$$\begin{aligned} \exists r.(F_i \sqcap V_i) &\sqsubseteq \overline{V}_i \quad \text{for all } i < n \\ \exists r.(F_i \sqcap \overline{V}_i) &\sqsubseteq V_i \quad \text{for all } i < n \end{aligned}$$

If a marker F_j for a different variable V_j is set, then V_i does not flip:

$$\begin{aligned} \exists r.(F_i \sqcap V_j) &\sqsubseteq V_j \quad \text{for all } i \leq n \text{ and } j < n \text{ with } i \neq j \\ \exists r.(F_i \sqcap \overline{V}_j) &\sqsubseteq \overline{V}_j \quad \text{for all } i < n \text{ and } j < n \text{ with } i \neq j \end{aligned}$$

Additionally, we would like to ensure that at least one of the F_i markers is true. This cannot be done in a straightforward way in $\mathcal{T}_G$. We will use the TBox $\mathcal{T}'_G$, which we define next. W.l.o.g., we assume that φ is in NNF. We first translate the formula φ into a set of GCIs as follows. For each $\psi \in \mathsf{sub}(\varphi)$, we introduce a concept name X_ψ. For each $\psi \in \mathsf{sub}(\varphi)$, we use $\sigma(\psi)$ to denote

- the concept name X_ψ if ψ is a non-literal and
- the concept name from $V_0, \ldots, V_{n-1}, \overline{V}_0, \ldots, \overline{V}_{n-1}$ corresponding to ψ if ψ is a literal.

Now we can translate each non-literal $\psi \in \mathsf{sub}(\varphi)$ into GCIs:

- if $\psi = \vartheta \wedge \chi$, then the GCI is $\sigma(\vartheta) \sqcap \sigma(\chi) \sqsubseteq X_\psi$;
- if $\psi = \vartheta \vee \chi$, then the GCIs are $\sigma(\vartheta) \sqsubseteq X_\psi$ and $\sigma(\chi) \sqsubseteq X_\psi$.

We introduce concept names $N, N', N'', N_0, \ldots, N_{n-1}$ that will be used as markers. Let k is the cardinality of Γ_1. First we add markers ensuring that (i) each variable has a truth value in every configuration, (ii) a least one of the flipping markers is set in every configuration, and (iii) the flipping marker denotes a variable controlled by the player whose turn it currently is:

$$\begin{aligned} V_i &\sqsubseteq N_i \quad \text{for all } i < n \\ \overline{V}_i &\sqsubseteq N_i \quad \text{for all } i < n \\ F_i &\sqsubseteq N' \quad \text{for all } i \in \{0, \ldots, k-1, n\} \\ F_i &\sqsubseteq N'' \quad \text{for all } i \in \{k, \ldots, n\} \end{aligned}$$

Next, we set a marker if P_2 has moved to reach a state in which φ is satisfied:

$$X_\varphi \sqcap P_2 \sqcap N'' \sqcap N_0 \sqcap \cdots \sqcap N_{n-1} \sqsubseteq N$$

Then, the marker N is pulled up inductively ensuring that if Player 1 is to move, there are the required $k+1$ successors, and if Player 2 is to move, there is the required single successor:

$$P_2 \sqcap N'' \sqcap N_0 \sqcap \cdots \sqcap N_{n-1} \sqcap \exists r.N \sqsubseteq N$$
$$P_1 \sqcap N' \sqcap N_0 \sqcap \cdots \sqcap N_{n-1} \sqcap \bigsqcap_{i \in \{0,\ldots,k-1,n\}} \exists r.(N \sqcap F_i) \sqsubseteq N$$

Finally, we require that P_1 moves first and that the initial configuration is labelled as described by Γ_I. Only if this is satisfied, the concept name B from $\mathcal{T}_G$ is implied:

$$P_2 \sqcap N \sqcap \bigsqcap_{i \in \Gamma_I} V_i \sqcap \bigsqcap_{i \notin \Gamma_I} \overline{V}_i \sqsubseteq B$$

Lemma 6. *Player 2 has a winning strategy in G iff $\mathcal{T}_G \cup \mathcal{T}'_G$ is not a conservative extension of $\mathcal{T}_G$.*

We have thus established the following result.

Theorem 3. *Deciding conservative extensions in $\mathcal{EL}$ is* ExpTime*-hard, thus* ExpTime*-complete.*

6 Model conservativity

In mathematical logic and software specification, there are (at least) two different kinds of conservative extensions. Until now, we have worked with the deductive version based on the consequence relation "$\models$". The second version is model-theoretic and defined as follows. 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 every model $\mathcal{I}$ of $\mathcal{T}_1$ can be extended to a model of $\mathcal{T}_1 \cup \mathcal{T}_2$ 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.

In the context of description logics, model conservative extensions have first been analyzed in [12], where it was argued that model conservative extensions are of interest for query answering modulo ontologies. For example, if $\mathcal{T}_1 \cup \mathcal{T}_2$ is a model conservative extension of $\mathcal{T}_1$, then the answers (over $\mathsf{sig}(\mathcal{T}_1)$) to a first-order query over $\mathsf{sig}(\mathcal{T}_1)$ given by $\mathcal{T}_1$ and $\mathcal{T}_1 \cup \mathcal{T}_2$ coincide. Indeed, the notion of a model conservative extension is more strict than the deductive 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 show the latter, let $\mathcal{T}_1 = \{A \sqsubseteq A\}$ and $\mathcal{T}_2 = \{\top \sqsubseteq \exists r.A\}$. It is not hard to see that $\mathcal{T}_1 \cup \mathcal{T}_2$ is a deductive conservative extension of $\mathcal{T}_1$ if $\mathcal{EL}$ (or even $\mathcal{ALC}$) is the assumed descripion logic, but it is not a model conservative extension.

Also in [12], it was shown that deciding model conservative extensions is undecidable and Π^1_1-complete in $\mathcal{ALC}$. In this section, we show the surprising result that model conservative extensions are undecidable even in $\mathcal{EL}$ (though we are not able to establish Π^1_1-hardness). The proof is by reduction of the halting problem for deterministic Turing machines on the empty tape. We assume w.l.o.g. that our Turing machines are such that the initial state is not reachable (directly or indirectly) from itself and that the halting state does not allow any further transitions. Let $M = (Q, \Sigma, \Gamma, \Delta, q_0, q_h)$ be a Turing machine. We construct TBoxes $\mathcal{T}_M$ and $\mathcal{T}'_M$ such that $\mathcal{T}_M \cup \mathcal{T}'_M$ is not a model conservative extension of $\mathcal{T}_M$ iff M halts on the empty tape. We use the following concept and role names for describing computations of M:

- the elements of Q and Γ as concept names;
- concept names head, before, and after to represent the relation of a tape cell to the head position;
- role names n (for *next tape cell*) and s (for *successor configuration*).

Our construction is such that models of $\mathcal{T}_M$ that cannot be extended to models of $\mathcal{T}'_M$ describe halting computations of M on the empty tape. Essentially, such models have the form of a grid, with the vertical edges labelled s and the horizontal ones labelled n. Thus, each row represents a configuration. We will enforce the roles n and s to be functional, except at row 0 and column 0 (because this does not seem possible). Therefore, the actual grid representing the computation of M starts at row 1 and column 1.

We start with the definition of $\mathcal{T}_M$. For now, it is easiest to simply assume n and s to be functional and confluent (which will be enforced later by $\mathcal{T}'_M$). We first set before and after correctly, exploiting the assumed functionality of n:

$$\exists n.\mathsf{before} \sqsubseteq \mathsf{before} \quad \exists n.\mathsf{head} \sqsubseteq \mathsf{before} \quad \mathsf{head} \sqsubseteq \exists n.\mathsf{after} \quad \mathsf{after} \sqsubseteq \exists n.\mathsf{after}.$$

Then we say that states are uniform over the tape: for all $q \in Q$,

$$q \sqsubseteq \exists n.q \quad \exists n.q \sqsubseteq q.$$

Exploiting that q_0 cannot reach itself and the above uniformity, we say that the tape is initially blank (where $b \in \Gamma$ is the blank symbol):

$$q_0 \sqsubseteq b.$$

For each transition $\delta(q, a) = (q', a', L)$, exploiting confluence of n and s, we set

$$\exists n.(q \sqcap \mathsf{head} \sqcap a) \sqsubseteq \exists s.(q' \sqcap \mathsf{head} \sqcap \exists n.a'),$$

and for each transition $\delta(q, a) = (q', a', R)$,

$$(q \sqcap \mathsf{head} \sqcap a) \sqsubseteq \exists s.(a' \sqcap q' \sqcap \exists n.\mathsf{head}).$$

We also say that symbols not under the head do not change: for all $a \in \Gamma$, put

$$a \sqcap \mathsf{before} \sqsubseteq \exists s.a, \;\; a \sqcap \mathsf{after} \sqsubseteq \exists s.a.$$

We would like to say that certain concept names such as before and head are disjoint. Since disjointness cannot be expressed in $\mathcal{EL}$, we revert to a trick that will become clear when $\mathcal{T}'_M$ is defined. For now, we introduce a concept name

D that serves as a marker for problems with disjointness: for all $q, q' \in Q$ with $q \neq q'$ and all $a, a' \in \Gamma$ with $a \neq a'$, put

$$q \sqcap q' \sqsubseteq D \quad a \sqcap a' \sqsubseteq D \quad \mathsf{before} \sqcap \mathsf{head} \sqsubseteq D \quad \mathsf{head} \sqcap \mathsf{after} \sqsubseteq D \quad \mathsf{before} \sqcap \mathsf{after} \sqsubseteq D.$$

Up to now, we simply have assumed the described grid structure, but we did not enforce it. In $\mathcal{T}_M$, we cannot do much more than saying that every point has the required successors:

$$\top \sqsubseteq \exists n.\top \sqcap \exists s.\top.$$

We now define $\mathcal{T}'_M$, introducing new atomic concepts N, A, B and a new role u. The concept name N serves as a marker. It is enforced to be true at the origin of the relevant part of the grid (point (1,1)) if the described computation reaches the halting state:

$$q_h \sqsubseteq N \quad \exists n.N \sqsubseteq N \quad \exists s.N \sqsubseteq N$$

It remains to ensure that a model $\mathcal{I}$ of $\mathcal{T}_M$ cannot be extended to a model of $\mathcal{T}'_M$ iff (i) r and s are functional, (ii) r and s are confluent, (iii) $D^{\mathcal{I}} = \emptyset$ (because then there are no problems with disjointness), (iv) the origin $(1,1)$ satisfies N (indicating that a halting state is reached), and (v) the described computation starts in the initial state with the head on the left-most cell and reaches the halting state. Suprisingly, all this can be achieved with two simple CIs:

$$\exists n.\exists s.(N \sqcap q_0 \sqcap \mathsf{head}) \sqsubseteq \exists u.(\exists n.\exists s.A \sqcap \exists s.\exists n.B)$$
$$A \sqcap B \sqsubseteq \exists u.D$$

Observe that any model $\mathcal{I}$ of $\mathcal{T}_M$ can indeed be extended to satisfy these additional CIs when any of the conditions (i) to (v) is violated, e.g., when D is non-empty or the roles n and s are functional anywhere except in row 0 and column 0. Conversely (and as shown in the proof of the following lemma), any model $\mathcal{I}$ of $\mathcal{T}_M$ that can be extended to these CIs violates one of (i) to (v).

Lemma 7. *$\mathcal{T}_M \cup \mathcal{T}'_M$ is not a model conservative extension of $\mathcal{T}_M$ iff M halts on the empty tape*

We have thus shown the following.

Theorem 4. *Deciding model conservative extensions in $\mathcal{EL}$ is undecidable.*

7 Conclusion

We have shown that deciding conservative extensions in $\mathcal{EL}$ is ExpTime-complete. As a next step, it is desirable to build on this foundation and design 'practical' algorithms. This is a serious challenge since conservative extensions are rather new as a reasoning problem and no experiences with implementing the associated algorithms have yet been made. (An exception is, of course, classical propositional logic, for which deciding conservative extensions corresponds to deciding the validy of quantified Boolean formulas of the form $\forall \boldsymbol{p} \exists \boldsymbol{q} \varphi(\boldsymbol{p}, \boldsymbol{q})$). The algorithm and results presented in this paper provide useful insights regarding crucial problems that have to be solve to develop a 'practical' procedure. For example, they indicate that such a procedure will rely on efficient algorithms for checking the existence of simulations between models.

References

1. G. Antoniou and K. Kehagias. A note on the refinement of ontologies. *Int. J. of Intelligent Systems*, 15:623–632, 2000.
2. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI'05)*, pages 364–369. Professional Book Center, 2005.
3. F. Baader, R. Küsters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI'99)*, pages 96–101. Morgan Kaufmann, 1999.
4. B. Bloom and R. Paige. Transformational design and implementation of a new efficient solution to the ready simulation problem. *Sci. Comput. Program.*, 24(3):189–220, 1995.
5. S. Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In *Proc. of the 16th European Conf. on Artificial Intelligence (ECAI-2004)*, pages 298–302. IOS Press, 2004.
6. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. DL-Lite: Tractable description logics for ontologies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, pages 602–607, 2005.
7. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 260–270, 2006.
8. E. Clarke and H. Schlingloff. Model checking. In *Handbook of Automated Reasoning*, volume II, chapter 24, pages 1635–1790. Elsevier Science, 2001.
9. S. Ghilardi, C. Lutz, and F. Wolter. Did I damage my ontology? A case for conservative extensions in description logics. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Repr. and Reasoning (KR'06)*, pages 187–197. AAAI Press, 2006.
10. B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. A logical framework for modularity of ontologies. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI'07)*. AAAI Press, 2007.
11. B. C. Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularity and web ontologies. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Repr. and Reasoning (KR'06)*, pages 198–209. AAAI Press, 2006.
12. C. Lutz, D. Walther, and F. Wolter. Conservative extensions in expressive description logics. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence IJCAI-07*. AAAI Press, 2007.
13. C. Lutz and F. Wolter. Conservative extensions in the lightweight description logic $\mathcal{EL}$. http://www.liv.csc.ac.uk/∼frank, 2007.
14. N. Sioutos, S. de Coronado, M. Haber, F. Hartel, W. Shaiu, and L. Wright. NCI thesaurus: a semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, 40(1):30–43, 2006.
15. K. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. 2000. Fall Symposium Special Issue.
16. L. J. Stockmeyer and A. K. Chandra. Provably difficult combinatorial games. *SIAM Journal on Computing*, 8(2):151–174, 1979.

A Omitted Proofs for Section 3

Lemma 1. Let $\mathcal{T}$ be a TBox and C a concept. For all $D \in \Delta^{C,\mathcal{T}}$ and all $E \in \mathsf{sub}(C) \cup \mathsf{sub}(\mathcal{T})$, we have $D \in E^{\mathcal{I}_{C,\mathcal{T}}}$ iff $\mathcal{T} \models D \sqsubseteq E$.

Proof. The proof is by induction on the structure of E. We first concentrate on the case when $E \in \mathsf{sub}(\mathcal{T})$. We only do the interesting case of the induction, i.e., $E = \exists r.F$.

"$\Rightarrow$". Let $D \in (\exists r.F)^{\mathcal{I}_{C,\mathcal{T}}}$. Then there is a $F' \in F^{\mathcal{I}_{C,\mathcal{T}}}$ with $(D, F') \in r^{\mathcal{I}_{C,\mathcal{T}}}$. We have $F' \in \mathsf{sub}(\mathcal{T})$ and can apply IH to the former, yielding $\mathcal{T} \models F' \sqsubseteq F$. Since $(D, F') \in r^{\mathcal{I}_{C,\mathcal{T}}}$, we have $\mathcal{T} \models D \sqsubseteq \exists r.F'$, thus $\mathcal{T} \models D \sqsubseteq \exists r.F$.

"$\Leftarrow$". Let $\mathcal{T} \models D \sqsubseteq \exists r.F$. Then $(D, F) \in r^{\mathcal{I}_{C,\mathcal{T}}}$. Since $\mathcal{T} \models F \sqsubseteq F$ and $F \in \mathsf{sub}(\mathcal{T})$, we get $F \in F^{\mathcal{I}_{C,\mathcal{T}}}$ from IH. Thus, $D \in (\exists r.F)^{\mathcal{I}_{C,\mathcal{T}}}$.

Now we consider the case where $E \in \mathsf{sub}(C) \setminus \mathsf{sub}(\mathcal{T})$, again doing only the interesting induction case where $E = \exists r.F$. The "$\Rightarrow$" is left to the reader.

"$\Leftarrow$". Let $\mathcal{T} \models D \sqsubseteq \exists r.F$. It suffices to show that $D \in D^{\mathcal{I}_{C,\mathcal{T}}}$: since we have already proved Lemma 1 for the case that $D \in \mathsf{sub}(\mathcal{T})$, $\mathcal{I}_{C,\mathcal{T}}$ is a model of $\mathcal{T}$; thus, $D \in D^{\mathcal{I}_{C,\mathcal{T}}}$ and $\mathcal{T} \models D \sqsubseteq \exists r.F$ implies $D \in (\exists r.F)^{\mathcal{I}_{C,\mathcal{T}}}$. Now, for $D \in \mathsf{sub}(\mathcal{T})$, $D \in D^{\mathcal{I}_{C,\mathcal{T}}}$ follows from $\mathcal{T} \models D \sqsubseteq D$ and the fact that $\mathcal{I}_{C,\mathcal{T}}$ is a model of $\mathcal{T}$. Now assume that $D \in \mathsf{sub}(C) \setminus \mathsf{sub}(\mathcal{T})$. Then D is of the form $F_0 \sqcap \bigsqcap_{(r,E_0) \in P} \exists r.E_0$ for a set P of pairs (r, E_0) and a conjunction F_0 of concept names. $D \in F_0^{\mathcal{I}_{C,\mathcal{T}}}$ follows from $\mathcal{T} \models D \sqsubseteq F_0$. By IH, $E_0 \in E_0^{\mathcal{I}_{C,\mathcal{T}}}$ for all $(r, E_0) \in P$. We also have $(D, E_0) \in r^{\mathcal{I}_{C,\mathcal{T}}}$ because $\exists r.E_0$ is a conjunct of D. Therefore $D \in (\exists r.E_0)^{\mathcal{I}_{C,\mathcal{T}}}$ for all $(r, E_0) \in P$. We obtain $D \in D^{\mathcal{I}_{C,\mathcal{T}}}$. ❑

Lemma 2. Let C, C_1, C_2, D be $\mathcal{EL}$-concepts and $\mathcal{T}$ a TBox. Then the following holds:

1. For all models $\mathcal{I}$ of $\mathcal{T}$ and all $d \in \Delta^{\mathcal{I}}$, the following conditions are equivalent:
   (a) $d \in C^{\mathcal{I}}$;
   (b) $(\mathcal{I}_{C,\mathcal{T}}, C) \leq (\mathcal{I}, d)$.
2. The following conditions are equivalent:
   (a) $\mathcal{T} \models C \sqsubseteq D$;
   (b) $C \in D^{\mathcal{I}_{C,\mathcal{T}}}$;
   (c) $(\mathcal{I}_{D,\mathcal{T}}, D) \leq (\mathcal{I}_{C,\mathcal{T}}, C)$.
3. If $\exists r.D \in (\mathsf{sub}(C_i) \cup \mathsf{sub}(\mathcal{T}))$ for all $i \in \{1, 2\}$, then $(\mathcal{I}_{C_1,\mathcal{T}}, D) \leq (\mathcal{I}_{C_2,\mathcal{T}}, D)$.

Proof. (1) The direction (b) $\Rightarrow$ (a) follows from Theorem 1 and the fact that $C \in C^{\mathcal{I}_{C,\mathcal{T}}}$. Conversely, let $\mathcal{I}$ be a model of $\mathcal{T}$ and $d \in C^{\mathcal{I}}$. Define a relation $S \subseteq \Delta^{\mathcal{I}_{C,\mathcal{T}}} \times \Delta^{\mathcal{I}}$ by setting $(D, e) \in S$ iff $e \in D^{\mathcal{I}}$. We show that S is a simulation. Let $(D, e) \in S$. Assume $D \in A^{\mathcal{I}_{C,\mathcal{T}}}$, with A a concept name. This implies $\mathcal{T} \models D \sqsubseteq A$, and $e \in A^{\mathcal{I}}$ follows from $e \in D^{\mathcal{I}}$ and $\mathcal{I} \models \mathcal{T}$. Now assume $(D, D') \in r^{\mathcal{I}_{C,\mathcal{T}}}$. Then $\mathcal{T} \models D \sqsubseteq \exists r.D'$ and we obtain $e \in (\exists r.D')^{\mathcal{I}}$. Hence, there exists $e' \in \Delta^{\mathcal{I}}$ with $(e, e') \in r^{\mathcal{I}}$ and $e' \in D'^{\mathcal{I}}$, which implies $(D', e') \in S$. It follows that S is a simulation. By definition, we have $(C, d) \in S$.

(2) Observe that the equivalence of (a) and (b) follows from Lemma 1 if $D \in \mathsf{sub}(C) \cup \mathsf{sub}(\mathcal{T})$. However, we consider arbitrary concepts D here. We show the following:

- (a) implies (b). Assume $\mathcal{T} \models C \sqsubseteq D$. Since $\mathcal{I}_{C,\mathcal{T}}$ is a model of $\mathcal{T}_1$ and $C \in C^{\mathcal{I}_{c,\mathcal{T}}}$, this implies $C \in D^{\mathcal{I}_{C,\mathcal{T}}}$.
- (b) implies (c). Immediate consequence of Point 1 of Lemma 2.
- (c) implies (a). Let $\mathcal{I}$ be a model of $\mathcal{T}$ and $d \in C^{\mathcal{I}}$. By Point 1 of Lemma 2, $(\mathcal{I}_{C,\mathcal{T}}, C) \leq (\mathcal{I}, d)$. Together with $(\mathcal{I}_{D,\mathcal{T}}, D) \leq (\mathcal{I}_{C,\mathcal{T}}, C)$ and transitivity of "$\leq$", we get $(\mathcal{I}_{D,\mathcal{T}}, D) \leq (\mathcal{I}, d)$. Again by Point 1 of Lemma 2, we obtain $d \in C^{\mathcal{I}}$.

(3) Let $\exists r.D \in (\mathsf{sub}(C_i) \cup \mathsf{sub}(\mathcal{T}))$, for all $i \in \{1, 2\}$. Then $D \in \Delta^{\mathcal{I}_{C_i,\mathcal{T}}}$, for all $i \in \{1, 2\}$. Define a relation $S \subseteq \Delta^{\mathcal{I}_{C_1,\mathcal{T}}} \times \Delta^{\mathcal{I}_{C_2,\mathcal{T}}}$ by setting $S := \{(E, E) \mid E \in \mathsf{sub}(D) \cup \mathsf{sub}(\mathcal{T})\}$. By construction, $(D, D) \in S$. It is easy to show that S is a simulation, hence $(\mathcal{I}_{C_1,\mathcal{T}}, D) \leq (\mathcal{I}_{C_2,\mathcal{T}}, D)$ as required. ❑

For the full proof of Lemma 3 we require two technical lemmas.

Lemma 8. *Suppose $\mathcal{T} \models C \sqsubseteq \exists r.D$. Then one of the following holds:*

- *there is a conjunct $\exists r.C'$ of C such that $\mathcal{T} \models C' \sqsubseteq D$;*
- *there is a $\exists r.C' \in sub(\mathcal{T})$ such that $\mathcal{T} \models C \sqsubseteq \exists r.C'$ and $\mathcal{T} \models C' \sqsubseteq D$.*

Proof. Let $\mathcal{T} \models C \sqsubseteq \exists r.D$. By Point 2 of Lemma 2, $C \in (\exists r.D)^{\mathcal{I}_{C,\mathcal{T}}}$. Thus, there is a $C' \in D^{\mathcal{I}_{C,\mathcal{T}}}$ such that $(C, C') \in r^{\mathcal{I}_{C,\mathcal{T}}}$. By definition of $\mathcal{I}_{C,\mathcal{T}}$, (i) $\exists r.C'$ is a conjunct of C or (ii) $\exists r.C' \in sub(\mathcal{T})$ and $\mathcal{T} \models C \sqsubseteq \exists r.C'$. It remains to argue that, in both cases, $\mathcal{T} \models C' \sqsubseteq D$. From $C' \in D^{\mathcal{I}_{C,\mathcal{T}}}$ and Point 1 of Lemma 2, we have $(\mathcal{I}_{D,\mathcal{T}}, D) \leq (\mathcal{I}_{C,\mathcal{T}}, C')$. By Point 3 of Lemma 2, we obtain $(\mathcal{I}_{C,\mathcal{T}}, C') \leq (\mathcal{I}_{C',\mathcal{T}}, C')$. By transitivity of $\leq$, $(\mathcal{I}_{D,\mathcal{T}}, D) \leq (\mathcal{I}_{C',\mathcal{T}}, C')$. From Point 2 of Lemma 2, we derive $\mathcal{T} \models C' \sqsubseteq D$ as desired. ❑

Lemma 9. *For all TBoxes $\mathcal{T}$ and concepts C, there is a concept D such that the following conditions are satisfied:*

1. *$\emptyset \models C \sqsubseteq D$,*
2. *$K_{\mathcal{T}}(C) = K_{\mathcal{T}}(D)$,*
3. *$|D| \leq |C|$,*
4. *the outdegree of D is bounded by $|\mathcal{T}|$.*

Proof. Let $\mathcal{T}$ be a TBox and C a concept. If the outdegree of C is bounded by $|\mathcal{T}|$, C itself is the wanted concept D. Assume that this is not the case. Then there exists a subconcept C_0 of C such that $C_0 = F \sqcap \bigsqcap_{(r,E) \in P} \exists r.E$, where F is a conjunction of concept names and $|P| > |\mathcal{T}|$. Let Q be a minimal subset of P such that for all $\exists r.G \in \mathsf{sub}(\mathcal{T})$, if there is a $(r, E) \in P$ with $\mathcal{T} \models E \sqsubseteq G$, then there is a $(r, E') \in Q$ with $\mathcal{T} \models E' \sqsubseteq G$. Clearly, the cardinality of Q is bounded by $|\mathcal{T}|$. Now, replace in C the subconcept C_0 with $C_1 := F \sqcap \bigsqcap_{(r,E) \in Q} \exists r.E$ and call the result C'. Clearly, $|C'| \leq |C|$. It is easy to construct a simulation from $(\mathcal{I}_{C,\emptyset}, C)$ to $(\mathcal{I}_{C',\emptyset}, C')$, and thus $\emptyset \models C \sqsubseteq C'$ by Point 2 of Lemma 2. In the following, we show that $K_\mathcal{T}(C) = K_\mathcal{T}(C')$. To obtain the desired concept D, it thus suffices to execute the described contraction information until the outdegree is bounded by $|\mathcal{T}|$.

"$\subseteq$". Immediate consequence of $\emptyset \models C \sqsubseteq C'$.

"$\supseteq$". Let $H \in \mathsf{sub}(\mathcal{T}) \setminus K_\mathcal{T}(C')$. We have to show that $H \notin K_\mathcal{T}(C)$. There is a model $\mathcal{I}$ of $\mathcal{T}$ with $d_0 \in C'^\mathcal{I} \setminus D^\mathcal{I}$. For each $(r, E) \in P \setminus Q$, take a copy $\mathcal{I}_{r,E}$ of the canonical model $\mathcal{I}_{E,\mathcal{T}}$ such that all these copies have disjoint domains, and their domains are disjoint from that of $\mathcal{I}$. Define a new interpretation $\mathcal{I}'$ as follows:

- take the disjoint union of $\mathcal{I}$ and the models $\mathcal{I}_{r,E}$, for all $(r, E) \in P \setminus Q$;
- for each $(r, E) \in P \setminus Q$ and each $d \in C_1^\mathcal{I}$, add the tuple $(d, d_{r,E})$ to $r^{\mathcal{I}'}$, where $d_{r,E} \in \Delta^{\mathcal{I}_{r,E}}$ was obtained from $E \in \Delta^{\mathcal{I}_{E,\mathcal{T}}}$ when taking a disjoint copy of $\mathcal{I}_{E,\mathcal{T}}$.

It is possible to prove the following by induction on the structure of D_0:

1. for all $(r, E) \in P \setminus Q$, all $d \in \Delta^{\mathcal{I}_{r,E}}$, and all concepts D_0, $d \in D_0^{\mathcal{I}'}$ iff $d \in D_0^{\mathcal{I}_{r,E}}$.
2. for all $d \in \Delta^\mathcal{I}$ and $D_0 \in \mathsf{sub}(\mathcal{T})$, $d \in D_0^\mathcal{I}$ iff $d \in D_0^{\mathcal{I}'}$.
3. for all $d \in \Delta^\mathcal{I}$ and $D_0 \in \mathsf{sub}(C')$, $d \in D_0^\mathcal{I}$ implies $d \in D_0^{\mathcal{I}'}$.

The only interesting case is in Point 2, when $D_0 = \exists r.D_0'$. Let $d \in (\exists r.D_0')^\mathcal{I}$. Then there is a $d' \in {D'}_0^\mathcal{I}$ such that $(d, d') \in r^\mathcal{I}$. It remains to note that $(d, d') \in r^{\mathcal{I}'}$ and apply IH. Conversely, let $d \in (\exists r.D_0')^{\mathcal{I}'}$. Then there is a $d' \in {D'}_0^{\mathcal{I}'}$ such that $(d, d') \in r^{\mathcal{I}'}$. If $d \in \Delta^\mathcal{I}$, we have $(d, d') \in r^\mathcal{I}$ and it again remains to apply IH. Now let $d' \in \mathcal{I}_{E,\mathcal{T}}$ for some $(r, E) \in P \setminus Q$. By Point 1 above, $d' \in {D'}_0^{\mathcal{I}'}$ implies $E \in {D'}_0^{\mathcal{I}_{E,\mathcal{T}}}$. With Point 2 of Lemma 2, we get $\mathcal{T} \models E \sqsubseteq D_0'$. By definition of Q, there is an $(r, E') \in Q$ such that $\mathcal{T} \models E' \sqsubseteq D_0'$. Since $(d, d') \in r^{\mathcal{I}'}$ with $d' \in \mathcal{I}_{E,\mathcal{T}}$, we have $d \in C_1^\mathcal{I}$ by construction of $\mathcal{I}'$. Thus, there is a $d'' \in (E')^\mathcal{I}$ with $(d, d'') \in r^\mathcal{I}$. Then $(d, d'') \in r^{\mathcal{I}'}$, $\mathcal{T} \models E' \sqsubseteq D_0'$ yields $d'' \in (D_0')^\mathcal{I}$, and we obtain $d'' \in (D_0')^{\mathcal{I}'}$ by IH.

Since $\mathcal{I}$ and all the $\mathcal{I}_{r,E}$ are models of $\mathcal{T}$ and by Points (1) and (2) above, it follows that $\mathcal{I}'$ is a model of $\mathcal{T}$. Point 2 implies that $d_0 \notin H^{\mathcal{I}'}$. By Point 3 , $d_0 \in C'^{\mathcal{I}'}$. We show that $d_0 \in (\exists r.E)^{\mathcal{I}'}$ for all $(r, E) \in P \setminus Q$. Together with $d_0 \in C'^{\mathcal{I}'}$, this implies $d_0 \in C^{\mathcal{I}'}$. Let $(r, E) \in P \setminus Q$. Since $d_{r,E} \in E^{\mathcal{I}_{r,E}}$, Point 1 yields $d_{r,E} \in E^{\mathcal{I}'}$. Together with $(d_0, d_{r,E}) \in r^{\mathcal{I}'}$, we get $d_0 \in (\exists r.E)^{\mathcal{I}'}$ as required. Thus, $d_0 \in C^{\mathcal{I}'} \setminus H^{\mathcal{I}'}$, implying $H \notin K_\mathcal{T}(C)$. ❑

Lemma 3. $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$ iff there exists a $\mathsf{sig}(\mathcal{T}_1)$-concept C and a $\mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$-concept $D \in \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2)$ such that

(a) $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq D$;

(b) $C \not\Rightarrow_1 D$;
(c) the outdegree of C is bounded by $|\mathcal{T}_1 \cup \mathcal{T}_2|$.

Proof. "⇒". Assume hat (a) to (c) are satisfied. By (b), there is a concept E with $\mathcal{T}_1 \cup \mathcal{T}_2 \models D \sqsubseteq E$ and $\mathcal{T}_1 \not\models C \sqsubseteq E$. From the former and (a), we get $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq E$, which implies that $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$.

"⇐". Assume that $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$. We first show only (a) and (b). If there is a counter-subsumption $C \sqsubseteq D$ with $D \in \mathsf{sub}(\mathcal{T}_1)$, then we are done: we have $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq D$ and $\mathcal{T}_1 \not\models C \sqsubseteq D$, therefore $C \not\Rightarrow_1 D$. Assume that no such counter-subsumption exists.

Let $C \sqsubseteq D$ be a counter-subsumption such that there is no counter-subsumption $C' \sqsubseteq D'$ with D' shorter than D. Then D is of the form $\exists r.D'$:

- If $D = \top$, then $\mathcal{T}_1 \models C \sqsubseteq D$, contradicting the fact that $C \sqsubseteq D$ is a counter-subsumption.
- If D is an atomic concept, then $D \in \mathsf{sub}(\mathcal{T}_1)$, which we have assumed not be the case.
- If D is a conjunction $D_1 \sqcap D_2$, then $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq D_i$ for all $i \in \{1,2\}$ and $\mathcal{T}_1 \not\models C \sqsubseteq D_i$ for some $i \in \{1,2\}$. Thus, one of $C \sqsubseteq D_1$ and $C \sqsubseteq D_2$ is a counter-subsumptionexample, contradicting the minimality of D.

By Lemma 8, $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq \exists r.D'$ implies that one of the following holds:

1. there exists a conjunct $\exists r.C'$ of C such that $\mathcal{T}_1 \cup \mathcal{T}_2 \models C' \sqsubseteq D'$;
2. there exists $\exists r.C' \in \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2)$ such that $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq \exists r.C'$ and $\mathcal{T}_1 \cup \mathcal{T}_2 \models C' \sqsubseteq D'$.

First note that, in the first case, we have $\mathcal{T}_1 \models C' \sqsubseteq D'$. For assume that the contrary holds. Since $\mathcal{T}_1 \cup \mathcal{T}_2 \models C' \sqsubseteq D'$, this implies that $C' \sqsubseteq D'$ is a counter-subsumption, contradicting the minimality of D. Also note that $\mathcal{T}_1 \models C' \sqsubseteq D'$ implies $\mathcal{T}_1 \models \exists r.C' \sqsubseteq \exists r.D'$.

Assume that Case 1 applies. Since $\exists r.C'$ is a conjunct of C, $\mathcal{T}_1 \models C \sqsubseteq \exists r.C'$. Together with $\mathcal{T}_1 \models \exists r.C' \sqsubseteq \exists r.D'$, we obtain $\mathcal{T}_1 \models C \sqsubseteq \exists r.D' = D$. It follows that $\mathcal{T}_1 \models C \sqsubseteq D$, contradicting the fact that $C \sqsubseteq D$ is a counter-subsumption.

Thus, Case 2 applies. We show that the concepts C and $\exists r.C'$ (substituted for D) satisfy Conditions (a) and (b). First, $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq \exists r.C'$ establishes Condition (a). For Condition (b), observe that $\mathcal{T}_1 \not\models C \sqsubseteq \exists r.D'$ and $\mathcal{T}_1 \cup \mathcal{T}_2 \models \exists r.C' \sqsubseteq \exists r.D'$. This means $C \not\Rightarrow_1 \exists r.C'$.

We now show how to additionally satisfy Condition (c). Let C be a $\mathsf{sig}(\mathcal{T}_1)$-concept and $D \in \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2)$ such that Points (a) and (b) hold. Take a concept C' that satisfies the four conditions from Lemma 9 (with $\mathcal{T}$ substituted by $\mathcal{T}_1 \cup \mathcal{T}_2$). By Condition 4, C' satisfies (c). By Condition 2, C' and D satisfy (a). Since $C \not\Rightarrow_1 D$, there is a $\mathsf{sig}(\mathcal{T}_1)$-concept E such that $\mathcal{T}_1 \cup \mathcal{T}_2 \models D \sqsubseteq E$ and $\mathcal{T}_1 \not\models C \sqsubseteq E$. By Condition 1, we have $\mathcal{T}_1 \models C \sqsubseteq C'$, and thus the latter implies $\mathcal{T}_1 \not\models C' \sqsubseteq E$. It follows that $C' \not\Rightarrow_1 D$, thus C' and D satisfy (b). ❑

B Omitted Proofs for Section 4

Lemma 10. *Let $\mathcal{T}$ be a TBox and $C = F_0 \sqcap \bigsqcap_{(r,E)\in P} \exists r.E$, where F_0 is a conjunction of concept names. Then*

$$K_{\mathcal{T}}(C) = K_{\mathcal{T}}(F_0 \sqcap \bigsqcap_{(r,E)\in P} \exists r.(\bigsqcap_{D\in K_{\mathcal{T}}(E)} D))$$

Proof. Let $C_0 = F_0 \sqcap \bigsqcap_{(r,E)\in P} \exists r.(\bigsqcap_{D\in K_{\mathcal{T}}(E)} D)$. We have to show that $K_{\mathcal{T}}(C) = K_{\mathcal{T}}(C_0)$.

"$\supseteq$". This follows from $\mathcal{T} \models C \sqsubseteq C_0$ (which follows from $\mathcal{T} \models E \sqsubseteq \bigsqcap_{D\in K_{\mathcal{T}}(E)} D$).

"$\subseteq$". Let $D \in \mathsf{sub}(\mathcal{T}) \setminus K_{\mathcal{T}}(C_0)$. Consider a model $\mathcal{I}$ of $\mathcal{T}$ with $d_0 \in C_0^{\mathcal{I}} \setminus H^{\mathcal{I}}$. For each $(r,E) \in P$, take a copy $\mathcal{I}_{r,E}$ of the canonical model $\mathcal{I}_{E,\mathcal{T}}$ such that all these copies have disjoint domains, and their domains are disjoint from that of $\mathcal{I}$. Define a new interpretation $\mathcal{I}'$ as follows:

- take the disjoint union of $\mathcal{I}$ and the models $\mathcal{I}_{r,E}$, for all $(r,E) \in P$;
- for each $(r,E) \in P$, add the tuple $(d_0, d_{r,E})$ to $r^{\mathcal{I}'}$, where $d_{r,E} \in \Delta^{\mathcal{I}_{r,E}}$ was obtained from $E \in \Delta^{\mathcal{I}_{E,\mathcal{T}}}$ when taking a disjoint copy of $\mathcal{I}_{E,\mathcal{T}}$.

It is not difficult to prove the following by induction on the structure of D_0:

1. for all $(r,E) \in P$, all $d \in \Delta^{\mathcal{I}_{r,E}}$, and all concepts D_0, $d \in D_0^{\mathcal{I}'}$ iff $d \in D_0^{\mathcal{I}_{r,E}}$.
2. for all $d \in \Delta^{\mathcal{I}}$ and $D_0 \in \mathsf{sub}(\mathcal{T})$, $d \in D_0^{\mathcal{I}}$ iff $d \in D_0^{\mathcal{I}'}$.

Since $\mathcal{I}$ and all the $\mathcal{I}_{r,E}$ are models of $\mathcal{T}$ and by Points (1) and (2) above, it follows that $\mathcal{I}'$ is a model of $\mathcal{T}$. Point 2 implies that $d_0 \notin H^{\mathcal{I}'}$. We show that $d_0 \in C^{\mathcal{I}'}$: by Point 2, $d_{r,E} \in E^{\mathcal{I}'}$ for all $(r,E) \in P$. Therefore $d_0 \in (\exists r.E)^{\mathcal{I}'}$ for all $(r,E) \in P$. Since we also have $d_0 \in F_0^{\mathcal{I}'}$, we obtain $d_0 \in C^{\mathcal{I}'}$. Thus, $d_0 \in C^{\mathcal{I}'} \setminus H^{\mathcal{I}'}$, implying $H \notin K_{\mathcal{T}}(C)$. ❑

C Omitted Proofs for Section 5

Before we can prove Lemma 6, we establish a technical lemma that asserts convexity of $\mathcal{EL}$. To state this property formally, let $\mathcal{EL}$ denote the extension of $\mathcal{EL}$ with a disjunction constructor $C \sqcup D$, with the obvious semantics $(C \sqcup D)^{\mathcal{I}} := C^{\mathcal{I}} \cup D^{\mathcal{I}}$.

Lemma 11. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox, and C, D, D' $\mathcal{EL}$-concepts. Then $\mathcal{T} \models C \sqsubseteq D \sqcup D'$ implies $\mathcal{T} \models C \sqsubseteq D$ implies or $\mathcal{T} \models C \sqsubseteq D'$.*

Proof. Let $\mathcal{T} \models C \sqsubseteq D \sqcup D'$. Take the model $\mathcal{I}_{C,\mathcal{T}}$ of $\mathcal{T}$. We have $C \in C^{\mathcal{I}_{C,\mathcal{T}}}$. Since $\mathcal{T} \models C \sqsubseteq D \sqcup D'$, we have $C \in D^{\mathcal{I}_{C,\mathcal{T}}}$ or $C \in D'^{\mathcal{I}_{C,\mathcal{T}}}$. By Point 2 of Lemma 2, this implies $\mathcal{T} \models C \sqsubseteq D$ or $\mathcal{T} \models C \sqsubseteq D'$ as required. ❑

As another preliminary, we make precise the notion of a *configuration* of the game G_4. A configuration of G is a pair (t, p) where t is a truth assignment for the variables in $\Gamma_1 \cup \Gamma_2$ and $p \in \{1, 2\}$ indicates the player that has moved to reach the current configuration. A *winning strategy for Player 2* is a finite tree (V, E, ℓ) where ℓ is a node labelling function that assigns to each node a configuration of G. The labelling is such that

1. the root is labelled with $(\Gamma_I, 2)$;
2. if a node is labelled with $(t, 1)$ (i.e., Player 2 is to move), then it has a single successor labelled $(t', 2)$, where t' is obtained from t by switching the truth value of at most one variable from Γ_2;
3. if a node is labelled with $(t, 2)$ (i.e., Player 1 is to move), then its successors are labelled $(t_0, 1), \ldots, (t_\ell, 2)$, where $t_0, \ldots, t_\ell$ are the configurations of G that can be obtained from t by switching the truth value of at most one variable from Γ_1;
4. if a leaf is labelled (t, i), then $i = 2$ and t satisfies φ.

We now come to the proof of Lemma 6.

Lemma 6. Player 2 has a winning strategy in G iff $\mathcal{T}_G \cup \mathcal{T}'_G$ is not a conservative extension of $\mathcal{T}_G$.

Proof. First assume that Player 2 has a winning strategy (V, E, ℓ) in G. We first define a mapping $m : V \to \{0, \ldots, n\}$ as follows: if $(v, v') \in E$, $\ell(v) = (t, i)$, and $\ell(v') = (t', i')$, then $m(v')$ is the variable that was switched to reach t' from t (we assume that $m(v') = n$ means that no variable was switched). If $v \in V$ is the root, $m(v) = n$ (this is arbitrary). We associate a concept $C(v)$ with each node $v \in V$:

$$C(v) := P_i \sqcap F_{m(v)} \sqcap \bigsqcap_{i \in t} V_i \sqcap \bigsqcap_{i \in (\Gamma_1 \cup \Gamma_2) \setminus t} \overline{V}_i$$

As a next step, we inductively associate another concept $W(v)$ with each node $v \in V$:

- if v is a leaf, then $W(v) := C(v)$;
- if v has successors $v_0, \ldots, v_{\ell-1}$, then $W(v) = C(v) \sqcap \bigsqcap_{i<\ell} \exists r.W(v_i)$.

Let ε be the root of (V, E, ℓ) and define $W := W(\varepsilon)$. It is not too difficult to verify that $\mathcal{T}_G \cup \mathcal{T}'_G \models W \sqsubseteq B$. We show that $\mathcal{T}_G \not\models W \sqsubseteq B$, and thus $\mathcal{T}_G \cup \mathcal{T}'_G$ is not a conservative extension of $\mathcal{T}_G$. Define a model $\mathcal{I}$ as follows:

- $\Delta^{\mathcal{I}} := \{W\} \cup \{C \mid \exists r.C \in \mathsf{sub}(W)\}$;
- $A^{\mathcal{I}} := \{C \in \Delta^{\mathcal{I}} \mid A \text{ is a conjunct in } C\}$;
- $r^{\mathcal{I}} := \{(C, C') \mid \exists r.C' \text{ is a conjunct in } C\}$.

It is easy to verify that $\mathcal{I}$ is a model of $\mathcal{T}_G$, and that $W \in W^{\mathcal{I}}$. Also, we have $B^{\mathcal{I}} := \emptyset$, and thus $\mathcal{T}_G \not\models W \sqsubseteq B$.

For the converse direction, our first step is to establish the following claim:

Claim. If $\mathcal{T}_G \cup \mathcal{T}'_G$ is not a conservative extension of $\mathcal{T}_G$, then there is a witness subsumption $C \sqsubseteq D$ with $D = B$.

Assume that $\mathcal{T}_G \cup \mathcal{T}'_G$ is not a conservative extension of $\mathcal{T}_G$ and let D be of minimal length such that there is a C with $\mathcal{T}_G \not\models C \sqsubseteq D$, but $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq D$. We first show that D is not of the form $D_1 \sqcap D_2$. Assume to the contrary that it is. Since $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq D$, we have $\mathcal{T}_G \cup \mathcal{T}'_G \models D_i$ for all $i \in \{1, 2\}$. Since $\mathcal{T}_G \not\models C \sqsubseteq D$, we have $\mathcal{T}_G \not\models C \sqsubseteq D_i$ for some $i \in \{1, 2\}$. Thus one of D_1 and D_2 is a counterexample against minimality of D.

Next, we show that D is not of the form $\exists r.D'$. Assume to the contrary that it is. By Lemma 8, there is a concept $\exists r.C' \in \mathsf{sub}(\mathcal{T}_G) \cup \mathsf{sub}(C)$ such that $\mathcal{T}_G \models C \sqsubseteq \exists r.C'$ and $\mathcal{T}_G \models C' \sqsubseteq D'$. Let D^* be D with the conjunct $\exists r.D'$ dropped. Then we have:

- $\mathcal{T}_G \not\models C \sqsubseteq D^*$: since $\mathcal{T}_G \not\models C \sqsubseteq D$, there is a model $\mathcal{I}$ of $\mathcal{T}_G$ and a $d \in (C \sqcap \neg D)^{\mathcal{I}}$. Then $d \in (\exists r.C')^{\mathcal{I}}$ and $\mathcal{T}_G \models C' \sqsubseteq D'$ implies $d \in (\exists r.D')^{\mathcal{I}}$. This together with $d \in \neg D^{\mathcal{I}}$ implies that there is a conjunct K of D such that $K \neq \exists r.D'$ and $d \notin K^{\mathcal{I}}$. Thus $d \in \neg(D^*)^{\mathcal{I}}$.
- $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq D^*$: implied by $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq D$.

Thus, we again have a contradiction against minimality of D.

At this point, D can only be a concept name or the $\top$ symbol. We can now finish the proof of the claim. Assume $B \neq D$. Take a model $\mathcal{I}$ of $\mathcal{T}_G$ and a $d \in \Delta^{\mathcal{I}}$ such that $d \in (C \sqcap \neg D)^{\mathcal{I}}$. Convert $\mathcal{I}$ into a model $\mathcal{I}'$ of $\mathcal{T}_G \cup \mathcal{T}'_G$ by

- defining the interpretation of the concept names introduced in $\mathcal{T}'_G$ such that all GCIs in $\mathcal{T}'_G$ except the last one are satisfied;
- then defining the interpretation of B such that $B^{\mathcal{I}} \subseteq B^{\mathcal{I}'}$ and the last GCI in $\mathcal{T}'_G$ is satisfied.

Since the extension of all concept and role names in $\mathcal{T}_G$ remained the same or got larger, $d \in C^{\mathcal{I}'}$. Since $D \neq B$ and B was the only concept name from $\mathcal{T}_G$ that we modified, $d \notin D^{\mathcal{I}'}$. Thus, $\mathcal{T}_G \cup \mathcal{T}'_G \not\models C \sqsubseteq D$, which is a contradiction.

Now that we know there exists a witness subsumption $C \sqsubseteq D$ with $D = B$, we can go about analyzing C. First observe that the only GCIs in $\mathcal{T}_G \cup \mathcal{T}'_G$ with B occurring on the right-hand side are

$$M \sqsubseteq B \text{ in } \mathcal{T}_G \quad \text{and} \quad P_2 \sqcap N \sqcap \bigsqcap_{i \in \Gamma_I} V_i \sqcap \bigsqcap_{i \notin \Gamma_I} \overline{V}_i \sqsubseteq B \text{ in } \mathcal{T}'_G,$$

and let us denote the left-hand side of the latter GCI with L. Since $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq B$, it follows that $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq M \sqcup L$: if this was not the case, there would be a model $\mathcal{I}$ of $\mathcal{T}_G \cup \mathcal{T}'_G$ and a $d \in (C \sqcup \neg M \sqcap \neg L)^{\mathcal{I}}$, and by modifying $\mathcal{I}$ by setting $B^{\mathcal{I}} := B^{\mathcal{I}} \setminus \{d\}$, we could obtain a model that contradicts $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq B$. By Lemma 11, $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq M \sqcup L$ implies $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq M$ or $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq L$. Since $M \sqsubseteq B$ is a GCI in $\mathcal{T}_G$, the former implies $\mathcal{T}_G \models C \sqsubseteq B$, which is a contradiction. Thus, $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq L$.

We now construct a winning strategy (V, E, ℓ) for Player 2. This is done by inductively constructing partial winning strategies $(V_0, E_0, \ell_0), (V_1, E_1, \ell_1), \dots$ such that for all $i > 0$, we have $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i$, where W'_i is the concept W induced by the winning strategy (V_i, E_i, ℓ_i) as defined in the left-to-right direction of this proof, with the modification that the conjunct N is added in $C(v)$.

To start, we set $V_0 := \{\varepsilon\}$, $E_0 := \emptyset$, and $\ell_0(\varepsilon) := (\Gamma_i, 2)$. Then W'_0 is the concept L from above and thus we have $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_0$ as required.

Now, let (V_i, E_i, ℓ_1) be already defined, and let v^* be a leaf with $\ell_i(v^*) = (t, p)$ such that t does not satisfy φ. The only GCIs in $\mathcal{T}_G \cup \mathcal{T}'_G$ with N occurring on the right-hand side are

$$\begin{aligned} X_\varphi \sqcap P_2 \sqcap N'' \sqcap N_0 \sqcap \cdots \sqcap N_{n-1} &\sqsubseteq N \\ P_2 \sqcap N'' \sqcap N_0 \sqcap \cdots \sqcap N_{n-1} \sqcap \exists r.N &\sqsubseteq N \\ P_1 \sqcap N' \sqcap N_0 \sqcap \cdots \sqcap N_{n-1} \sqcap \bigsqcap_{i \in \{0,\dots,k-1,n\}} \exists r.(N \sqcap F_i) &\sqsubseteq N \end{aligned}$$

Let us denote the left-hand sides with L_1, L_2, and L_3, respectively. For all concepts D, we use $W'_i[D]$ to denote the modification of W'_i obtained by redefining $C(v)$ specifially for the node v^* by including the conjunct D. Since (i) $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i$, (ii) $C(v^*)$ includes N as a conjunct, and (iii) the above listed GCIs are the only ones in $\mathcal{T}_G \cup \mathcal{T}'_G$ with N on the right-hand side, we have $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[L_1 \sqcup L_2 \sqcup L_3]$. By Lemma 11, it follows that $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[L_j]$, for some $j \in \{1, 2, 3\}$. To continue, we make a case distinction according to whether $p = 1$ or $p = 2$, starting with $p = 2$.

We first show that $\mathcal{T}_G \cup \mathcal{T}'_G \not\models C \sqsubseteq W'_i[L_1]$ and $\mathcal{T}_G \cup \mathcal{T}'_G \not\models C \sqsubseteq W'_i[L_3]$:

- Assume $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[L_1]$. Since $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i$, we have

$$\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[\bigsqcap_{i \in t} V_i \sqcap \bigsqcap_{i \in \{0,\dots,n-1\} \setminus t} \overline{V}_i]. \qquad (*)$$

  Using the disjointness GCIs in $\mathcal{T}_G$, it is not hard to show that there is only one truth assignment with Property $(*)$. More specifically, if any truth assignment $t' \neq t$ satisfies $(*)$, then there is a $k < n$ with

$$\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[V_k \sqcap \overline{V}_k]$$

  and we can use the GCIs in $\mathcal{T}_G$ to show that this implies $\mathcal{T}_G \models C \sqsubseteq B$, a contradiction. Since X_φ is a conjunct in L_1, we have $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[X_\varphi]$. Analyzing the GCIs derived from the formula φ and exploiting that t is the unique truth assignment satisfying $(*)$, we can show that t satisfies φ. This is a contradiction to our choice of φ^*.
- Assume $\mathcal{T}_G \cup \mathcal{T}'_G \not\models C \sqsubseteq W'_i[L_3]$. Then $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[P_1]$. Since $C(v^*)$ includes the conjunct P_2, we have $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[P_1 \sqcap P_2]$, implying $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq \exists r^m.(P_1 \sqcap P_2)$ for some $m \geq 0$. It is easy to use the GCIs in $\mathcal{T}_G$ to show that this implies $\mathcal{T}_G \models C \sqsubseteq B$, a contradiction.

Thus $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[L_2]$. By analyzing L_2 and the GCIs carefully in a similar way as we have done above, it is not hard to show that $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[F_\ell]$ for some $\ell \in \{k, \ldots, n\}$. Also, we can show that $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[\exists r.N]$. This, in turn, means that we have $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[\exists r.L_{j'}]$, for some $j' \in \{1, 2, 3\}$. The cases $j' = 1$ and $j' = 2$ can be ruled out: since both L_1 and L_2 include the conjunct P_2 and so does $C(v^*)$, we could again argue that then $\mathcal{T}_G \models C \sqsubseteq B$ (using the GCI $\exists r.P_2 \sqsubseteq P_1$), which is a contradiction. Thus, $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[\exists r.L_3]$ implying $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[\exists r.N']$. By analyzing the GCIs in $\mathcal{T}_G \cup \mathcal{T}'_G$ with N' on the right-hand side, we have $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[\exists r.F_k]$ for some $k \in \{0, \ldots, k-1, n\}$. Using the disjointness GCIs in $\mathcal{T}_G$, we can show that k is unique (otherwise $\mathcal{T}_G \models C \sqsubseteq B$). Define the truth assignment t' as t with the value of the variable V_k flipped if $k < n$. Otherwise, $t' = t$. Then set $V_{i+1} := V_i \cup \{v\}$, $E_{i+1} := E_i \cup (v^*, v)$ and set $\ell_{i+1}(v) = (t', 1)$.

We have to show that $\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_{i+1}$, which boils down to showing that

$$\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_{i+1}[\bigsqcap_{i \in t'} V_i \sqcap \bigsqcap_{i \in \{0,\ldots,n-1\} \setminus t'} \overline{V}_i],$$

where $W'_{i+1}[D]$ denotes the modification of W'_{i+1} obtained by redefining $C(v)$ specifially for the node v (not v^*!) by including the conjunct D. Since $C(v)$ includes the conjunct N and all GCIs in $\mathcal{T}_G \cup \mathcal{T}'_G$ with N on the right-hand side have $N_0, \ldots, N_{n-1}$ as conjuncts on the left-hand side, we can analyze the GCIs in $\mathcal{T}'_G$ to show that there is a truth assignment t'' such that

$$\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_{i+1}[\bigsqcap_{i \in t''} V_i \sqcap \bigsqcap_{i \in \{0,\ldots,n-1\} \setminus t''} \overline{V}_i].$$

Using the disjointness GCIs in $\mathcal{T}_G$, we can show that t'' is unique. Argueing as above, we can strengthen the last statement to

$$\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_{i+1}[F_k \sqcap \bigsqcap_{i \in t''} V_i \sqcap \bigsqcap_{i \in \{0,\ldots,n-1\} \setminus t''} \overline{V}_i].$$

where k is the value that has to be flipped to reach t' from t. We can use the GCIs in $\mathcal{T}_G$ axiomatizing the behaviour of the F_i markers to show that

$$\mathcal{T}_G \cup \mathcal{T}'_G \models C \sqsubseteq W'_i[\bigsqcap_{i \in t^*} V_i \sqcap \bigsqcap_{i \in \{0,\ldots,n-1\} \setminus t^*} \overline{V}_i],$$

where t^* is obtained from t'' by flipping the value of k. Since it follows from the disjointment statements in $\mathcal{T}_G$ that there is only a single t^* with this property, it follows that $t^* = t$, and thus $t' = t''$.

This finishes the case where $p = 2$. The case $p = 1$ is very similar, and details are left to the reader.

It remains to argue that the construction of the partial winning strategies eventually terminates, yielding a complete winning strategy. If the construction does not terminate, we have $\mathcal{T}_G \cup \mathcal{T}'_G \models \exists r^m.\top$ for all $m \geq 0$. It is easy to prove that this is impossible, since no GCI in $\mathcal{T}_G \cup \mathcal{T}'_G$ has an existential restriction of the right-hand side.

❏

D Omitted Proofs for Section 6

Lemma 7. $\mathcal{T}_M \cup T'_M$ is not a model conservative extension of T_M iff M halts on the empty tape

Proof. "⇐" Assume that M does not halt on the empty tape and let $c_0, \dots, c_k$ be the halting computation of M. Extend this computation to an infinite sequence of computations by setting $c_\ell := c_k$ for all $\ell > k$. We define an interpretation $\mathcal{I}$ as follows:

- $\Delta^\mathcal{I} := \mathbb{N} \times \mathbb{N}$;
- $s^\mathcal{I} := \{((i,j),(i+1,j)) \mid i,j \geq 0\}$;
- $n^\mathcal{I} := \{((i,j),(i,j+1)) \mid i,j \geq 0\}$;
- $q^\mathcal{I} := \{(i,j) \mid i,j > 0$ and the state in c_{j-1} is $q\}$ for all $q \in Q$;
- $a^\mathcal{I} := \{(i,j) \mid i,j > 0$ and tape cell $i-1$ in c_{j-1} is labelled $a\}$ for all $a \in \Gamma$;
- $\mathsf{head}^\mathcal{I} := \{(i,j) \mid i,j > 0$ and the head position in c_{j-1} is $j-1\}$;
- $\mathsf{before}^\mathcal{I} := \{(i,j) \mid (i',j) \in \mathsf{head}^\mathcal{I}$ for some $i' > i\}$;
- $\mathsf{after}^\mathcal{I} := \{(i,j) \mid (i',j) \in \mathsf{head}^\mathcal{I}$ for some $i' < i\}$;
- $D^\mathcal{I} = \emptyset$.

It is not hard to verify that $\mathcal{I}$ is a model of $\mathcal{T}_M$ (setting $c_\ell = c_k$ for all $\ell > k$ is justified by the fact that M does not allow any transitions in the halting state). Moreover, $\mathcal{I}$ cannot be extended to a model of $\mathcal{T}_M \cup \mathcal{T}'_M$: we have $(0,0) \in (\exists n.\exists s.(N \sqcap q_0 \sqcap \mathsf{head}))^\mathcal{I}$, so we have to interpret u, A, and B such that $(0,0) \in (\exists u.(\exists n.\exists s.A \sqcap \exists s.\exists n.B))^\mathcal{I}$. To do this, we have to interpret A and B such that $(i,j) \in (A \sqcap B)^\mathcal{I}$ for some $i,j \geq 0$. To obtain a model of $\mathcal{T}'_M$, we must then ensure that $(i,j) \in D^\mathcal{I}$. This, however, is impossible since $D^\mathcal{I} = \emptyset$ is fixed. It follows that $\mathcal{T}_M \cup \mathcal{T}'_M$ is not a conservative extension of $\mathcal{T}_M$.

"⇒". Assume that M does not halt on the empty tape and let $\mathcal{I}$ be a model of $\mathcal{T}_M$. We have to show that $\mathcal{I}$ can be extended to a model of $\mathcal{T}_M \cup \mathcal{T}'_M$. If $q_h^\mathcal{I} = \emptyset$, then we simply set $A^\mathcal{I} := B^\mathcal{I} := N^\mathcal{I} := u^\mathcal{I} := \emptyset$. If $q_h^\mathcal{I} \neq \emptyset$, let $N^\mathcal{I}$ be the smallest set such that $q_h^\mathcal{I} \subseteq N^\mathcal{I}$, $(\exists n.N)^\mathcal{I} \subseteq N^\mathcal{I}$, and $(\exists s.N)^\mathcal{I} \subseteq N^\mathcal{I}$. If the result is such that $(\exists n.\exists s.(N \sqcap q_0 \sqcap \mathsf{head}))^\mathcal{I} = \emptyset$, we are done. So assume the contrary. First assume that

(i) There are $d, d_1, d_2, d_3, d_4 \in \Delta^\mathcal{I}$ with $d n^\mathcal{I} d_1 s^\mathcal{I} d_2$ and $d s^\mathcal{I} d_3 n^\mathcal{I} d_4$ such that $d_2 \neq d_4$.

Then we can set $u^\mathcal{I} := \Delta^\mathcal{I} \times \{d\}$, $A^\mathcal{I} := \{d_2\}$, and $B^\mathcal{I} := \{d_4\}$, and obtain a model of $\mathcal{T}'_M$. Now assume

(ii) There are $d_1, d_2, d_3, d_4 \in \Delta^\mathcal{I}$ with $d_1 n^\mathcal{I} d_2 s^\mathcal{I} d_4$, $d_1 s^\mathcal{I} d_3 n^\mathcal{I} d_4$, and $d_4 \in C^\mathcal{I}$.

Then we can set $u^{\mathcal{I}} := \Delta^{\mathcal{I}} \times \{d_1\}$ and $A^{\mathcal{I}} := B^{\mathcal{I}} := \{d_4\}$ to obtain a model of $\mathcal{T}'_M$. Now assume that neither (i) nor (ii) are the case. We show that this is impossible since it implies that M halts on the empty tape. Let $d_0 \in (\exists n.\exists s.(N \sqcap q_0 \sqcap \mathsf{head}))^{\mathcal{I}}$. Then there is a $d'_0 \in \Delta^{\mathcal{I}}$ and a $d \in (N \sqcap q_0 \sqcap \mathsf{head})^{\mathcal{I}}$ such that $d_0 n^{\mathcal{I}} d'_0 s^{\mathcal{I}} d$. For $d' \in \Delta^{\mathcal{I}}$, we say that d' is *reachable* from d *in n steps* if there exists a sequence $d_0, \ldots, d_n$ with $d_0 = d$, $d_n = d'$, and $(d_i, d_{i+1}) \in n^{\mathcal{I}} \cup s^{\mathcal{I}}$ for all $i < n$. We say that that d' is *reachable* from d if d' is reachable from d in n steps, for some $n \geq 0$. We first show the following:

Claim. Let d' be reachable from d. Then we have:

1. there are $d_1, d_2, d_3 \in \Delta^{\mathcal{I}}$ such that $d_1 n^{\mathcal{I}} d_2 s^{\mathcal{I}} d'$ and $d_1 s^{\mathcal{I}} d_3 n^{\mathcal{I}} d'$;
2. if $d' n^{\mathcal{I}} e$ and $d' n^{\mathcal{I}} e'$, then $e = e'$;
3. if $d' s^{\mathcal{I}} e$ and $d' s^{\mathcal{I}} e'$, then $e = e'$;

Point 1 is proved by induction on the minimal n such that d' is reachable from d in n steps. For the induction start, we have $d' = d$. Recall that $d_0 n^{\mathcal{I}} d'_0 s^{\mathcal{I}} d$. By the CIs in $\mathcal{T}_M$, there are $d_1, d_2 \in \Delta^{\mathcal{I}}$ such that $d_0 s^{\mathcal{I}} d_1 n^{\mathcal{I}} d_2$. Since (i) does not hold, $d_2 = d$ and we are done. For the induction step, let d' be reachable from d in $n > 0$ steps. Then there is a d_1 such that d_1 is reachable from d in $n-1$ steps and $d_1 n^{\mathcal{I}} d'$ or $d_1 s^{\mathcal{I}} d'$. We only treat the first case since the second is analogous. By IH, there is a d_2 such that $d_2 s^{\mathcal{I}} d_1$. By the CIs in $\mathcal{T}_M$, there are d_3 and d_4 such that $d_2 n^{\mathcal{I}} d_3 s^{\mathcal{I}} d_4$. Since (i) does not holds, $d_4 = d'$ and we are done.

Now for Points 2 and 3. We only treat Point 2 explicitly since Point 3 is analogous. Let d' be reachable from d and let $e, e' \in \Delta^{\mathcal{I}}$ such that $d' n^{\mathcal{I}} e$ and $d' n^{\mathcal{I}} e'$. By Point 1, there is a d_1 such that $d_1 s^{\mathcal{I}} d'$. By the CIs in $\mathcal{T}_M$, there are d_2, d_3 such that $d_1 n^{\mathcal{I}} d_2 s^{\mathcal{I}} d_3$. Since (i) does not hold, we have $d_3 = e = e'$, and are done. This finishes the proof of the claim.

Set $R := \{d' \in \Delta^{\mathcal{I}} \mid d' \text{ is reachable from } d\}$. Points 2 and 3 of the claim together with the fact that (i) does not hold implies that we can easily find a bijection $\tau : R \rightarrow \mathbb{N} \times \mathbb{N}$ such that for all $e, e' \in R$, we have

– $e n^{\mathcal{I}} e'$ iff $\tau(e) = (i, j)$ and $\tau(e) = (i+1, j)$ for some $i, j \in \mathbb{N}$;
– $e s^{\mathcal{I}} e'$ iff $\tau(e) = (i, j)$ and $\tau(e) = (i, j+1)$ for some $i, j \in \mathbb{N}$.

Our aim is to read off a halting computation from M on the empty tape from $\mathcal{I}$, being guided by τ. To do this, we first show that (a) for all $q, q' \in Q$ with $q \neq q'$, $q^{\mathcal{I}} \cap q'^{\mathcal{I}} \cap R = \emptyset$, (b) for all $a, a' \in \Sigma$ with $a \neq a'$, $a^{\mathcal{I}} \cap a'^{\mathcal{I}} \cap R = \emptyset$, and (c) $\mathsf{before}^{\mathcal{I}} \cap R$, $\mathsf{after}^{\mathcal{I}} \cap R$, and $\mathsf{head}^{\mathcal{I}} \cap R$ are pairwise disjoint. Since the argument is the same in all three cases, we concentrate on (a). Assume $e \in q^{\mathcal{I}} \cap q'^{\mathcal{I}} \cap R$. By the GCIs in $\mathcal{T}_M$, $d' \in C^{\mathcal{I}}$. By Point 1 of the claim, there are $d_1, d_2, d_3 \in \Delta^{\mathcal{I}}$ such that $d_1 n^{\mathcal{I}} d_2 s^{\mathcal{I}} d'$ and $d_1 s^{\mathcal{I}} d_3 n^{\mathcal{I}} d'$. This is a contradiction to the fact that (ii) is false

We can now read off a halting computation from M in the obvious way: the i-th configuration is described by the elements $R_i := \{d \in R \mid \tau(d) = (j, i) \text{ for some } j \geq 0\}$. By the CIs in $\mathcal{T}_M$ and (a), there is a unique state $q \in Q$

such that $R_i \subseteq q^{\mathcal{I}}$. By the CIs in $\mathcal{T}_M$ and (b), for each $j \geq 0$, there is a unique $a \in \Sigma$ such that $\tau^{-1}(j,i) \in a^{\mathcal{I}}$. And by the CIs in $\mathcal{T}_M$ and (c), there is a unique $j \geq 0$ such that $\tau^{-1}(j,i) \in \mathsf{head}^{\mathcal{I}}$. Let us call the resulting sequence of configurations $c_0, c_1, \ldots$. By choice of d above and the CIs in $\mathcal{T}_M$, c_0 is the initial configuration of M on the empty tape. By the CIs in $\mathcal{T}_M$, c_{i+1} is a successor configuration of c_i for all $i \geq 0$. By definition of $N^{\mathcal{I}}$ and since $d \in N^{\mathcal{I}}$, it follows that we eventually reach a halting configuration. ❑