The Combined Complexity of Reasoning with Closed Predicates in Description Logics*

Nhung Ngo[1], Magdalena Ortiz[2], and Mantas Šimkus[2]

[1] Free University of Bozen-Bolzano
ngo@inf.unibz.it
[2] Vienna University of Technology
ortiz@kr.tuwien.ac.at|simkus@dbai.tuwien.ac.at

Abstract. We investigate the computational cost of combining the open and closed world assumptions in Description Logics. Unlike previous works, which have considered data complexity and focused mostly on lightweight DLs, we study combined complexity for a wide range of DLs, from lightweight to very expressive ones. From existing results for the standard setting, where all predicates are interpreted under the open world assumption, and the well-known relationship between closed predicates and concept constructors like disjunction and nominals, we infer bounds on the combined complexity of reasoning in the presence of closed predicates. We show that standard reasoning requires exponential time even for weak logics like $\mathcal{EL}$, while answering conjunctive queries becomes hard at least for coNExpTime, and in most cases even for 2ExpTime. An important stepping stone for our results, that is interesting on its own right, is to prove that conjunctive query answering in (plain) $\mathcal{ALCO}$ is hard for coNExpTime in combined complexity. This singles out nominals as a previously unidentified source of additional complexity when answering queries over expressive DLs.

1 Introduction

As fragments of classical first-order predicate logic, description logics (DLs) have an *open-world* semantics. That is, knowledge bases (KBs) are interpreted as the set of *all* relational structures that satisfy what is explicitly stated in the KB, and where any statement whose truth is not directly implied by the knowledge base can be interpreted in an arbitrary way. However, this open-world view of DLs is not the most adequate in all cases, and in particular, when DLs are used to describe domain-specific knowledge to be leveraged when querying data sources, but the sources stem from traditional (closed-world) databases that fully describe the instances that are to be included in a relation. For example, when the students enrolled in some course are extracted from a database, this information should be considered complete, and query answering algorithms should exploit this to exclude irrelevant models and infer more query answers.

Combining open and closed world reasoning is not a new topic in DLs [3], but it has received renewed attention in recent years [20,19,7,30]. A prominent proposal for achieving partial closed world reasoning is to use DBoxes instead of ABoxes as the

* This work has been supported by the Austrian Science Fund (FWF) projects T515 and P25207.

assertional component of KBs [30]. Syntactically, a DBox looks just like an ABox, but semantically, it is interpreted like a database: the instances of the concepts and roles in the DBox are given exactly by the assertions it contains, and the *unique name assumption* is made for the *active domain* of the individuals occurring in it. We follow a recent generalization of this setting, where instead of replacing ABoxes by DBoxes, we enrich the terminological component of KBs with a set of concepts and roles that are to be interpreted as *closed predicates* [19]. In this way, some ABox assertions are interpreted under closed semantics, as in DBoxes, while others are considered open, as in ABoxes.

There are not many works studying the complexity of reasoning with closed predicates. For the DL-Lite family and for $\mathcal{EL}$, the *data complexity* of *ontology mediated query answering* has been considered [7,19]. The authors of these works consider a conjunctive query together with a terminological component (a TBox and possibly a set of closed predicates), and study the complexity of answering such a query over an input data instance (an ABox or a DBox). Under the standard open-world semantics for all predicates, this is a central problem that has received great attention in the last decade in the DL community. Most research focuses on the cases where the problem is tractable, or even first-order rewritable. Unfortunately, the tractability of query answering is lost in the presence of DBoxes, even for the core fragments of DL-Lite and $\mathcal{EL}$ [7]. In a nutshell, closed predicates cause the *convexity* property to be lost, allowing a KB to entail a disjunction of facts without entailing any of the disjuncts. An in-depth analysis of this and a careful classification of tractable cases can be found in [19].

In this paper, we take a closer look at the computational complexity of reasoning in the presence of closed predicates. Unlike previous works, we consider the *combined complexity* of reasoning, that is, not only the data is considered as an input, but also the terminological information, and in the case of query answering, the query. Rather than focusing on a few lightweight DLs, we consider a range of logics including very expressive ones, and use existing results in the literature to infer many tight bounds on the computational complexity of query answering. It was shown already in [30] that closed predicates can be simulated, under the standard open world semantics, in any extension of $\mathcal{ALC}$ that supports nominals, and conversely, nominals can be simulated by closed predicates (the latter does not depend on any of the availability of any the constructs of $\mathcal{ALC}$). It is also easy to show that closed predicates suffice to easily express full disjunction and atomic negation in any logic supporting qualified existential restrictions, hence adding closed predicates to plain $\mathcal{EL}$ already results in the full expressiveness of $\mathcal{ALCO}$, and makes standard reasoning require exponential time in the worst case.

For query answering, we build on the reduction from $\mathcal{ALC}$ to $\mathcal{EL}$ to show that the constructors that make query answering 2EXPTIME-hard in extensions of $\mathcal{ALC}$ (namely inverses [17], or transitive roles together with role hierarchies [6]), have the same effect in the analogous extensions of $\mathcal{EL}$ in the presence of closed predicates (i.e., $\mathcal{ELI}$ and $\mathcal{ELH}^{\mathsf{trans}}$). However, since the precise complexity of query answering in $\mathcal{ALCO}$ remains open, we cannot infer tight bounds for the extensions of $\mathcal{EL}$ with closed predicates that do not support these additional constructs. This leads us to the main technical contribution of the paper: we show that conjunctive query answering over $\mathcal{ALCO}$ (with the standard open-world semantics) is coNEXPTIME-hard. Hence the same holds in the presence of closed predicates for $\mathcal{EL}$ and its extensions. Although we leave a matching upper bound open for future work, we exhibit nominals (or closed predicates) as a previously unidentified source of increased complexity for query answering in expressive DLs.

2 Preliminaries

We assume the reader is familiar with DLs and, in particular, with $\mathcal{EL}$ and $\mathcal{ALCO}$. We use N_C and N_R for concept names and roles, respectively. The notions of a *TBox* $\mathcal{T}$, an *ABox* $\mathcal{A}$, and an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ are defined in the usual way. The notions of satisfaction $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$ are also as usual. We make the standard name assumption (SNA), i.e., $a^{\mathcal{I}} = a$ for all $\mathcal{I}$ and individuals a. For combining the open- and closed-world semantics, we enrich KBs with a set Σ of *closed predicates*. That is, we consider *knowledge bases (KBs)* $\mathcal{K} = (\mathcal{T}, \Sigma, \mathcal{A})$, where $\mathcal{T}$ is a TBox, $\Sigma \subseteq N_C \cup N_R$, and $\mathcal{A}$ is an ABox. We call Σ the set of *closed predicates* in $\mathcal{K}$. For such a $\mathcal{K}$ and an interpretation $\mathcal{I}$, we write $\mathcal{I} \models \mathcal{K}$ if
(a) $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$,
(b) for all concept names $A \in \Sigma$, if $e \in A^{\mathcal{I}}$, then $A(e) \in \mathcal{A}$, and
(c) for all roles $r \in \Sigma$, if $(e, e') \in r^{\mathcal{I}}$, then $r(e, e') \in \mathcal{A}$.
In case $\Sigma = \emptyset$, $\mathcal{K}$ is boils down to a usual DL KB and $\mathcal{I} \models \mathcal{K}$ captures the usual notion of satisfaction. In this case, we may simply write $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.

Note that in this paper KBs with closed predicates have the semantics as in [19], which relies on the SNA. However, all complexity results of this paper can be recast for the semantics of [7] that employs a weaker form of unique name assumption.

3 Standard Reasoning

Interpreting some predicates as closed allows one to simulate negation, disjunction, and nominals in plain $\mathcal{EL}$, making it as expressive as full $\mathcal{ALCO}$.

Theorem 1. *Assume a consistent $\mathcal{ALCO}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. For every nominal $\{a\}$ that appears in $\mathcal{K}$, let D_a be a fresh concept name. Then we can construct in linear time an $\mathcal{EL}$ KB $\mathcal{K}' = (\mathcal{T}', \Sigma, \mathcal{A}')$ with closed predicates such that:*

1. *Every model $\mathcal{I}$ of $\mathcal{K}'$ is a model of $\mathcal{K}$. Moreover, $D_a^{\mathcal{I}} = \{a\}^{\mathcal{I}}$ for every $\{a\}$ in $\mathcal{K}$.*
2. *Every model $\mathcal{I}$ of $\mathcal{K}$ can be transformed into a model of $\mathcal{K}'$ by modifying the interpretation of concept names and roles that do not appear in $\mathcal{K}$.*

Proof. The extension of $\mathcal{EL}$ with atomic negation (i.e., negation is applied to concept names only), denoted $\mathcal{EL}^{\neg}$, is known to be a notational variant of $\mathcal{ALC}$ [1]. Hence we simply show how to construct $\mathcal{K}' = (\mathcal{T}', \Sigma, \mathcal{A}')$ for a given $\mathcal{ELO}^{\neg}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. We do this in two steps: first we eliminate nominals and obtain from $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ an $\mathcal{EL}^{\neg}$ KB $\mathcal{K}_1 = (\mathcal{T}_1, \Sigma_1, \mathcal{A}_1)$. Then we transform $\mathcal{K}_1 = (\mathcal{T}_1, \Sigma_1, \mathcal{A}_1)$ into the desired $\mathcal{K}' = (\mathcal{T}', \Sigma, \mathcal{A}')$ in $\mathcal{EL}$. Let

$$\Sigma_1 = \{D_a \mid \{a\} \text{ appears in } \mathcal{K}\}, \qquad \mathcal{A}_1 = \mathcal{A} \cup \{D_a(a) \mid \{a\} \text{ appears in } \mathcal{K}\}.$$

This ensures that $D_a^{\mathcal{I}} = \{a\}^{\mathcal{I}}$ in every model of $\mathcal{A}_1$ where the predicates in Σ_1 are interpreted as closed. Hence we can simply replace each concept $\{a\}$ by D_a in $\mathcal{T}$ to obtain the desired $\mathcal{T}_1$. Next, for eliminating negation from $(\mathcal{T}_1, \Sigma_1, \mathcal{A}_1)$, we let

$\Sigma = \Sigma_1 \cup \{D_\bot, D_1, D_2, D_u\}$ $\qquad$ $\mathcal{A}' = \mathcal{A}_1 \cup \{D_u(a_1), D_u(a_2), D_1(a_1), D_2(a_2)\}$

To obtain $\mathcal{T}'$ from $\mathcal{T}_1$, we replace every negated concept name $\neg A$ by a fresh concept name $\bar{A}$, and add the following axioms, where r_A is a fresh role name:

$$A \sqcap \bar{A} \sqsubseteq D_\bot \quad (1) \qquad \exists r_A.D_1 \sqsubseteq A \quad (3)$$

$$\top \sqsubseteq \exists r_A.D_u \quad (2) \qquad \exists r_A.D_2 \sqsubseteq \bar{A} \quad (4)$$

Note that, since there are no assertions of the form $D_\bot(a)$ in $\mathcal{A}'$, we have $D_\bot^\mathcal{I} = \emptyset$ in every model $\mathcal{I}$ of $\mathcal{K}'$, and hence $D_\bot$ behaves as the special concept $\bot$. This and axiom (1) ensure disjointness of A and $\bar{A}$, while axioms (2) – (4) together with the assertions for the closed predicates D_u, D_1 and D_2 ensure that $e \in A^\mathcal{I}$ or $e \in (\bar{A})^\mathcal{I}$ for every $e \in \Delta^\mathcal{I}$. Indeed, let $\mathcal{I}$ be a model of $\mathcal{K}'$ and let $e \in \Delta^\mathcal{I}$ be arbitrary. Then by axiom (2) there exists $e' \in \Delta^\mathcal{I}$ such that $(e, e') \in r^\mathcal{I}$ and $e' \in D_u^\mathcal{I}$. But since D_u is closed, then $e' \in \{a_1, a_2\}$. If $e' = a_1$, then $e' \in D_1^\mathcal{I}$, so $e \in (\exists r_A.D_1)^\mathcal{I}$ and $e \in A^\mathcal{I}$ by axiom (3). Analogously, if $e' = a_2$, we can use axiom (4) to infer that $e \in (\bar{A})^\mathcal{I}$. This ensures that $\bar{A}$ has exactly the same extension as $\neg A$ in every model of $\mathcal{K}$, and the claim follows.

This easy reduction allows us to extend to $\mathcal{EL}$ hardness results known for $\mathcal{ALC}$. We can also infer upper bounds for logics with closed predicates, from known results under the standard open-world semantics, by exploiting the fact that in DLs that contain $\mathcal{ALC}$, closed predicates can be simulated using nominals (see [26] and Prop. 1 in [30]):

Theorem 2. *For every DL KB $(\mathcal{T}, \Sigma, \mathcal{A})$ there exists a logically equivalent KB of the form $(\mathcal{T} \cup \mathcal{T}', \emptyset, \mathcal{A})$, where $\mathcal{T}'$ is a set of $\mathcal{ALCO}$ axioms whose size is polynomially bounded by the size of $\mathcal{A}$.*

This already allows us to obtain an almost complete picture of the landscape for standard reasoning tasks. The following theorem is stated for KB satisfiability, but note that it also applies to other traditional reasoning tasks like subsumption and instance checking since they are mutually reducible to each other in $\mathcal{ALC}$, and hence also in any logic containing $\mathcal{EL}$ with closed predicates.

Corollary 1. *The following bounds for deciding satisfiability of $(\mathcal{T}, \Sigma, \mathcal{A})$ hold:*

1. EXPTIME-*complete if $\mathcal{T}$ and $\mathcal{A}$ are in any DL containing $\mathcal{EL}$ and contained in $\mathcal{SHOQ}$ or $\mathcal{SHOI}$.*
2. NEXPTIME-*complete if $\mathcal{T}$ and $\mathcal{A}$ are in any DL containing $\mathcal{ELIF}$ and contained in $\mathcal{SHOIQ}$.*
3. N2EXPTIME-*complete if $\mathcal{T}$ and $\mathcal{A}$ are in $\mathcal{SRIQ}$ or $\mathcal{SROIQ}$, and in* 2EXPTIME *if $\mathcal{T}$ and $\mathcal{A}$ are in $\mathcal{SROQ}$ or $\mathcal{SROI}$.*

Proof. The lower bound of item (1) follows from Theorem 1 and the well known EXPTIME-hardness of KB satisfiability in $\mathcal{ALC}$ [29]. Similarly, the hardness of items (2) and (3) follows from Theorem 1 together with the NEXPTIME-hardness of $\mathcal{ALCOIF}$ [31], and the N2EXPTIME-hardness of $\mathcal{SROIQ}$ [13]. For the upper bounds, we use Theorem 2 and the fact that KB satisfiability is decidable in the mentioned bounds for the listed logics: $\mathcal{SHOQ}$ and $\mathcal{SHOI}$ in EXPTIME, $\mathcal{SROQ}$ and $\mathcal{SROI}$ in 2EXPTIME, $\mathcal{SHOIQ}$ in NEXPTIME, $\mathcal{SROIQ}$ in N2EXPTIME (see [4,13] and references therein).

4 Query Answering

In this section we consider the query answering problem over DL KBs. We cannot easily transfer complexity upper bounds from known KB satisfiability since, in general, queries are not naturally expressible in the syntax of DLs, and encoding them as part of a KB usually requires exponential space. We focus in *conjunctive queries (CQs)*, whose syntax and semantics are defined in the usual way. In a nutshell, a CQ is a conjunction of atoms of the forms $A(x)$ or $r(x, y)$, for a concept name A or a role name r, and variables x, y. In what follows, all queries are *Boolean queries* with all variables existentially quantified. The decision problem we consider is *query entailment*: deciding whether a given query q is true in all the models of a given KB $(\mathcal{T}, \Sigma, \mathcal{A})$.

It is well known that, under the classical open-world semantics, CQ entailment is hard for 2EXPTIME in most expressive DLs, but the complexity drops to EXPTIME in Horn fragments that disallow disjunction. Unfortunately, since the presence of closed predicates causes disjunction to be expressible, 2EXPTIME-hardness extends to many extensions of $\mathcal{EL}$. For CQs, 2EXPTIME-hardness can be shown whenever the DL supports inverse roles [17], or a single left-identity axiom $r \circ t \sqsubseteq t$, or a transitive super role of some role [6]. If we consider query languages that extend CQs, like positive queries or (fragments of) conjunctive (2-way) regular path queries (C(2)RPQs), the same hardness holds already for plain $\mathcal{ALC}$ [25], and hence for $\mathcal{EL}$ with closed predicates.

Below we denote by $\mathcal{EL}^{\mathsf{LI}}$ the extension of $\mathcal{EL}$ with a single left-identity axiom $r \circ t \sqsubseteq t$, and by $\mathcal{ELH}^{\mathsf{trans}}$ the extension of $\mathcal{EL}$ with role inclusions and a transitive role. Note that both logics are sublogics of $\mathcal{EL}^{++}$.

Theorem 3. *Deciding $(\mathcal{T}, \Sigma, \mathcal{A}) \models q$ is hard for 2EXPTIME in all the following cases:*

1. *$\mathcal{T}$ and $\mathcal{A}$ are in $\mathcal{ELI}$ and q is a CQ.*
2. *$\mathcal{T}$ and $\mathcal{A}$ are in $\mathcal{EL}^{\mathsf{LI}}$ or in $\mathcal{ELH}^{\mathsf{trans}}$, and q is a CQ.*
3. *$\mathcal{T}$ and $\mathcal{A}$ are in $\mathcal{EL}$ and q is either a positive query, a $*$-free CRPQ, or a $*$-free C2RPQ with only two variables.*

Proof. We have shown in Theorem 1 that for every $\mathcal{ALC}$ KB $\mathcal{K}$ there is an $\mathcal{EL}$ KB $\mathcal{K}'$ that has essentially the same models, and may only differ in the interpretation of symbols not occurring in $\mathcal{K}$. Hence, for every query q, we have $\mathcal{K} \models q$ iff $\mathcal{K}' \models q$. This translation can be applied to extensions of $\mathcal{ALC}$, and results in a KB with the same properties in the analogous extension of $\mathcal{EL}$. In particular, an $\mathcal{ALCI}$ KB is rewritten into a $\mathcal{ELI}$ one, and an $\mathcal{SH}$ KB into an $\mathcal{ELH}^{\mathsf{trans}}$ one. From this an existing results for $\mathcal{ALC}$ and its extensions, we obtain the desired lower bounds: item 1 follows from [18], item 2 follows from [6], and item 3 follows from [25].

Matching upper bounds are known, even for significantly more expressive queries and logics: in the standard setting, with no closed predicates, entailment of *positive two-way regular path queries (P2RPQs)* is in 2EXPTIME for any DL contained in $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$, $\mathcal{SHIQ}$, $\mathcal{SHOQ}$, or $\mathcal{SHOI}$ [4]. From this and Theorem 2, we get the same upper bound for $\mathcal{ZOQ}$, $\mathcal{ZOI}$, $\mathcal{SHOQ}$, $\mathcal{SHOI}$ and their sublogics.

Corollary 2. *Let q be a P2RPQ. Then deciding $(\mathcal{T}, \Sigma, \mathcal{A}) \models q$ is* 2EXPTIME *complete for $\mathcal{T}$ and $\mathcal{A}$ in any DL containing $\mathcal{EL}$ and contained in $\mathcal{ZOQ}$, $\mathcal{ZOI}$, $\mathcal{SHOQ}$, $\mathcal{SHOI}$. The same holds for q a CQ if $\mathcal{T}$ and $\mathcal{A}$ are in a DL containing $\mathcal{ELI}$ or $\mathcal{EL}^{\mathsf{LI}}$.*

Corollary 2 implies that query entailment in the presence of closed predicates is almost always 2EXPTIME-complete in combined complexity. But there are some exceptions. On the one hand, the interaction of nominals, inverses, and counting makes query entailment very challenging. In the plain open-world setting, entailment of conjunctive queries is coN2EXPTIME-hard for $\mathcal{ALCOIF}$ [10], and it has been shown to be decidable [28], but no elementary upper bounds on its complexity are known. For its extension with transitive roles, and for the well known $\mathcal{SHOIQ}$, decidability remains open. Hence, in the presence of closed predicates, we do not get any interesting upper bounds for DLs that simultaneously support inverses and counting, like $\mathcal{ALCIF}$ and $\mathcal{SHIQ}$. Moreover, obtaining such bounds seems very hard. We remark that the authors of [7] proved that query entailment in $\mathcal{ALCIF}$ with closed predicates and query entailment under the standard open-world semantics in $\mathcal{ALCOIF}$ are mutually reducible.

On the other hand, the mentioned 2EXPTIME lower bounds for CQs require the presence of either inverse roles, left identity axioms, or transitivity and role hierarchies. For $\mathcal{EL}$ (with closed predicates), $\mathcal{ALC}$, and their extensions that have neither inverses nor left identities, we only have the EXPTIME lower bound from KB satisfiability, and the 2EXPTIME upper bound of Corollary 2. Without closed predicates, CQ entailment for plain $\mathcal{ALC}$, and even for $\mathcal{ALCHQ}$, is feasible in single exponential time [18,24]. A natural question is whether nominals, or equivalently, closed predicates, can be added to $\mathcal{ALCHQ}$ without increasing the worst-case complexity of CQ entailment. Unfortunately, the answer is negative (unless coNEXPTIME = EXPTIME), as we show next.

A coNEXPTIME lower bound for CQ entailment in $\mathcal{ALCO}$

In this section, we show that deciding whether $(\mathcal{T}, \mathcal{A}) \not\models q$ for a given CQ q and a given $\mathcal{ALCO}$ KB $(\mathcal{T}, \mathcal{A})$ (with the standard open-world semantics), is hard for non-deterministic single exponential time. By Theorem 1, the same applies to $\mathcal{EL}$ in the presence of closed predicates.

Before we start with the proof, we recall a useful property of $\mathcal{ALCO}$: for query answering, it is enough to focus on *forest-shaped models*. A *forest* is a set F of non-empty words such that $w \cdot c \in F$ with w non-empty implies $w \in F$. An interpretation $\mathcal{I}$ is *forest-shaped* if there is a bijection f from its domain to a forest, such that

- $f(a^{\mathcal{I}})$ has length one for every individual a, and
- $(e, e') \in r^{\mathcal{I}}$ implies that either $e' = a$ for some individual a, or $f(e')$ is of the form $f(e) \cdot c$ for some symbol c.

For many DLs and query languages, it has been shown that query entailment can be decided over forest shaped interpretations. This applies also to CQs over $\mathcal{ALCO}$ KBs:

Lemma 1 ([9,4]). *Let $\mathcal{K}$ be a given $\mathcal{ALCO}$ KB and let q be a CQ. Then $\mathcal{K} \not\models q$ iff there is a forest shaped interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \not\models q$.*

Now we show our lower bound. The proof is by reduction from the following coNEXPTIME-complete variant of the tiling problem:

Definition 1 (Domino System). *A* domino system $\mathfrak{D}$ *is a triple* (T, H, V)*, where* $T = \{0, \ldots, k-1\}$*,* $k \geq 0$*, is a finite set of* tile types *and* $H, V \subseteq T \times T$ *represent the* horizontal and vertical matching conditions*. Let* $\mathfrak{D}$ *be a domino system and* $c = c_0, \ldots, c_{n-1}$ *an* initial condition*, i.e. an* n*-tuple of tile types. A mapping* $\tau : \{0, \ldots, 2^{n+1}-1\} \times \{0, \ldots, 2^{n+1}-1\} \to T$ *is a* solution *for* $\mathfrak{D}$ *and* c *iff for all* $x, y < 2^{n+1}$*, the following holds (where* $\oplus_i$ *denotes addition modulo* i*):*

- *if* $\tau(x, y) = t$ *and* $\tau(x \oplus_{2^{n+1}} 1, y) = t'$*, then* $(t, t') \in H$
- *if* $\tau(x, y) = t$ *and* $\tau(x, y \oplus_{2^{n+1}} 1) = t'$*, then* $(t, t') \in V$
- $\tau(i, 0) = c_i$ *for* $i < n$*.*

For the reduction, we do not need a full $\mathcal{ALCO}$ knowledge base, but a simple ABox with single concept assertion $C_{\mathfrak{D},c}(a)$ for a complex $\mathcal{ALCO}$ concept $C_{\mathfrak{D},c}$. We show how to translate a given domino system $\mathfrak{D}$ and initial condition $c = c_0 \cdots c_{n-1}$ into an assertion $C_{\mathfrak{D},c}(a)$ and query $q_{\mathfrak{D},c}$ such that each forest-shaped model $\mathcal{I}$ of $C_{\mathfrak{D},c}(a)$ that satisfies $\mathcal{I} \not\models q_{\mathfrak{D},c}$ encodes a solution to $\mathfrak{D}$ and c, and conversely each solution to $\mathfrak{D}$ and c gives rise to a model of $C_{\mathfrak{D},c}(a)$ with $\mathcal{I} \not\models q_{\mathfrak{D},c}$.

Our reduction is based on the proof of coNExpTime-hardness of *rooted* query entailment in $\mathcal{ALCI}$ [17], and also resembles the similar proof for $\mathcal{S}$ [6]. In fact, the first part of the concept $C_{\mathfrak{D},c}$, which generates forest models that encode a potential solutions, is essentially as in [17]. The second part and the query are quite different, since they exploit nominals to detect errors in potential solutions.

Constructing the ABox. We now define the complex concept $C_{\mathfrak{D},c}$, and the desired ABox is $\{C_{\mathfrak{D},c}(a)\}$. We assume that $C_{\mathfrak{D},c}$ is a conjunction of the form $C^1_{\mathfrak{D},c} \sqcap C^2_{\mathfrak{D},c}$.

For convenience, let $m = 2n+2$. The purpose of the first conjunct $C^1_{\mathfrak{D},c}$ is to enforce a binary tree of depth m, whose edges are labeled with a single role r, and whose leaves are labeled with the numbers $0, \ldots, 2^m - 1$ of a binary counter C, implemented using concept names $B_0, \ldots, B_m$. Intuitively, each of these leaves ℓ stores a position in the $2^{n+1} \times 2^{n+1}$-grid to be tiled: the bits $B_0, \ldots B_n$ encode the horizontal position x, and the bits $B_{n+1}, \ldots B_m$ encode the vertical position y. We also use a concept name D_i for each tile type $i \in T$. Each leaf g storing a position (x, y) has as r-children three 'grid nodes' g_h, g_{right}, and g_{up} labeled G, which satisfy all the following conditions:

1. g_h represents the grid node with position (x, y), and stores the same bit address as g (that is, g_h and g coincide on the interpretation of all B_i).
2. g_{right} and g_{up} represent the right- and up-neighbor of g, and respectively store the addresses $(x \oplus_{2^{n+1}} 1, y)$ and $(x, y \oplus_{2^{n+1}} 1)$.
3. g_h is labeled G_h, while g_{right} and g_{up} are labeled G_s.
4. g_h (resp., g_{right}, g_{up}) satisfies exactly one concept D_i, representing the assigned tile type $\tau(x, y)$ (resp., $\tau(x \oplus_{2^{n+1}} 1, y)$, $\tau(x, y \oplus_{2^{n+1}} 1)$).
5. The tiling of the g_h nodes respects the initial condition, that is, if ℓ_h stores the position $(i, 0)$, then it satisfies D_{c_i}.
6. The tiling of g_{right} and g_{up} respect the matching conditions, that is, if g_h satisfies D_i, g_{right} satisfies D_j, and g_{up} satisfies $D_{j'}$, then $(D_i, D_j) \in H$ and $(D_i, D_{j'}) \in V$.

The tree we have described almost describes a solution for $\mathfrak{D}$, except for the crucial fact that different copies of the same node in the grid may have different types assigned.

That is, for an address (x, y), the g_{right} and g_{up} nodes with address (x, y) need not satisfy the same D_i as the g_h with address (x, y). We call a model $\mathcal{I}$ of $C^1_{\mathfrak{D},c}(a)$ *proper* if it satisfies the following condition:

($\star$) For every pair $g \in G^{\mathcal{I}}_h$, $g' \in G^{\mathcal{I}}_s$ such that $g \in B_i$ iff $g' \in B_i$ for all $0 \leq i \leq m$, there exists some $i < k$ such that $\{g, g'\} \subseteq D^{\mathcal{I}}_i$.

We can use an $\mathcal{ALC}$ concept $C^1_{\mathfrak{D},c}$ to enforce a tree as above, such that deciding the existence of a solution for $\mathfrak{D}$ and c reduces to finding a proper model of $C^1_{\mathfrak{D},c}$. Such constructions exist in the literature, and in fact, the concept we described is just a minor modification of the conjunction $C^1_{\mathfrak{D},c} \sqcap \cdots \sqcap C^6_{\mathfrak{D},c}$ given in [17], hence we omit its rather technical definition. Instead, we rely on the following claim:

Lemma 2 (implicit in [17]). *Let $\mathfrak{D}$ be a domino system and c an initial condition. Then we can build an $\mathcal{ALC}$ concept $C^1_{\mathfrak{D},c}$ such that there exists a solution for $\mathfrak{D}$ and c iff there exists a proper model of $C^1_{\mathfrak{D},c}(a)$. Moreover, the size of $C^1_{\mathfrak{D},c}$ and the time needed to construct it are polynomially bounded by the size of $\mathfrak{D}$.*

We construct below a query $q_{\mathfrak{D},c}$ that does *not* match a forest model of $C^1_{\mathfrak{D},c}(a)$ iff ($\star$) is satisfied. By Lemmas 2 and 1, this suffices to decide whether there exists a solution for $\mathfrak{D}$ and c. But before defining $q_{\mathfrak{D},c}$, we define the second conjunct $C^2_{\mathfrak{D},c}$ of $C_{\mathfrak{D},c}$. Its purpose is to add nodes and labels to the forest models of $C^1_{\mathfrak{D},c}(a)$ that allow us to test for ($\star$) using a CQ.

For defining $C^2_{\mathfrak{D},c}$, we use the following additional alphabet symbols:

- two individual names a_i and $\bar{a}_i$ and one concept name A_i for each bit B_i, $0 \leq i \leq m$,
- an individual name t_j for each tile type $j < k$,
- a concept T stating that some individual stands for a tile type.

Each G node g is linked via r-arcs to the individuals $a_i, \bar{a}_i$ that encode its bit address. We also link g nodes to the individuals that stand for the tile types, but we do it differently for the G_h nodes and the G_s nodes, as follows:

- If g is a G_h-node with tile type D_i, then g has an r-arc to t_i.
- If g is a G_s-node with tile type D_i, then g has an r-arc to each t_j with $j \neq i$.

Finally, we make both a_i and $\bar{a}_i$ instances of A_i, for each bit i, and we make all tile types t_j instances of T. Formally, this is all ensured using the conjunction $C^2_1 \sqcap C^2_2$ of the following two concepts:

$$C^2_1 = \forall r^{m+1}.\big(\prod_{0 \leq i \leq m} \big(\; B_i \rightarrow \exists r.\{a_i\} \sqcap \neg B_i \rightarrow \exists r.\{\bar{a}_i\} \sqcap$$
$$\prod_{0 \leq i < k} D_i \rightarrow \big((G_h \rightarrow \exists r.\{t_i\}) \sqcap (G_s \rightarrow (\prod_{0 \leq j < k, j \neq i} \exists r.t_j)) \,\big)$$

$$C^2_2 = \forall r^{m+2}.\big((\prod_{0 \leq i \leq m} \{a_i, \bar{a}_i\} \rightarrow A_i \big) \sqcap \big(\{t_0, \ldots t_{k-1}\} \rightarrow T \big) \big)$$

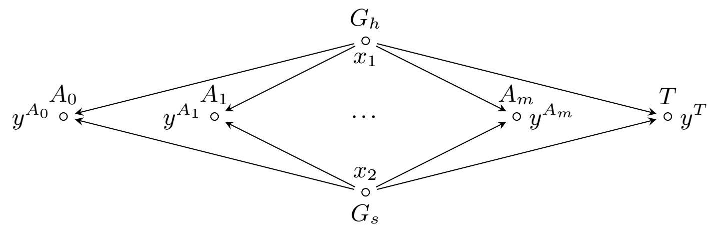

Fig. 1: The query $q_{\mathfrak{D},c}$

Now we are ready to define our ABox $\{C_{\mathfrak{D},c}(a)\}$, by taking $C^2_{\mathfrak{D},c} = C^2_1 \sqcap C^2_2$ as defined above, $C^1_{\mathfrak{D},c}$ as in Lemma 2, and $C_{\mathfrak{D},c} = C^1_{\mathfrak{D},c} \sqcap C^2_{\mathfrak{D},c}$. Every model of $C_{\mathfrak{D},c}(a)$ is a model of $C^1_{\mathfrak{D},c}(a)$, and every forest model of $C^1_{\mathfrak{D},c}(a)$ can be extended to a model of $C_{\mathfrak{D},c}(a)$ while preserving properness, hence from Lemma 2 we get:

Lemma 3. *$\mathfrak{D}$ and c have a solution iff there exists a proper forest model of $C_{\mathfrak{D},c}(a)$.*

It is only left to define a query $q_{\mathfrak{D},c}$ that matches a forest model of $C_{\mathfrak{D},c}(a)$ if and only if it is not proper. We will rely on the following properties ensured by $C^1_{\mathfrak{D},c}$. First, the connections to the individuals representing the bit address ensure the following:

(†) Let g, g' be two G-nodes. Then g and g' share an r-arc to a common individual from $a_i, \bar{a}_i$ for each $0 \leq i \leq m$ iff they have the same bit address.

Since the links to the tile types for the G_h-nodes and for the G_s-nodes are inverted, we also have:

(‡) Let g_h be a G_h-node and g_s be a G_s-node. Then there exists some t_j such that both g_h and g_s have an r-arc to t_j iff g_h and g_s have different tile types.

Hence, to establish non-properness it suffices to find a G_h-node and a G_s-node that share an r-arc to a common individual from $a_i, \bar{a}_i$ for each $0 \leq i \leq m$ (and hence share the same address), but also share a link to a t_j node (and hence have different tile type). This is done with the following query:

$$\begin{aligned}q_{\mathfrak{D},c} = &\exists x_1, x_2, y^{A_0}, , \dots, y^{A_m}, y^T . G_h(x_1), G_s(x_2),\\ &r(x_1, y^{A_0}), A_0(y^{A_0}), \dots, r(x_1, y^{A_m}), A_m(y^{A_m}), r(x_1, y^T), T(y^T),\\ &r(x_2, y^{A_0}), A_0(y^{A_0}), \dots, r(x_2, y^{A_m}), A_m(y^{A_m}), r(x_2, y^T), T(y^T).\end{aligned}$$

The query $q_{\mathfrak{D},c}$ is illustrated in Figure 1. To see that $q_{\mathfrak{D},c}$ has a match iff ($\star$) fails, we first note that x_1 can only be matched to a G_h node and x_2 to a G_s node. Each y^{A_i} must be matched to an instance of A_i, which is one of a_i and $\bar{a}_i$. Since x_1 and x_2 are connected to all the y^{A_i}, then they need to have either a_i or $\bar{a}_i$ as common successor, for each i, in every model where the query has a match. By (†), this is the case exactly when they have the same bit address. The variable y^T can match instances of T, which are only the tile type individuals t_j, and since x_1 and x_2 are both connected to y^T, we have that x_1 and x_2 can only be matched to nodes sharing a link to a common t_j. By (‡), and since the matches of x_1 and x_2 are a G_h node and a G_s node, they must have a different tile type. Hence the query has a match iff there are a G_h node g_h and a G_s node g_s that have the same bit address and different tile types, that is, there is a pair violating the condition of ($\star$) and the model is not proper. Hence we get that there is a model $\mathcal{I}$ of $C_{\mathfrak{D},c}(a)$ where there is no match for $q_{\mathfrak{D},c}$, and iff there exist a proper model of $C_{\mathfrak{D},c}(a)$ iff there is a solution for $\mathfrak{D}$ and c.

Lemma 4. *$C_{\mathfrak{D},c}(a) \not\models q_{\mathfrak{D},c}$ iff there is a solution for $\mathfrak{D}$ and c.*

From this, the hardness of the given tiling problem, and Theorem 1, we get:

Theorem 4. *The following problems are hard for* coNEXPTIME*:*

- *deciding $(\emptyset, \{C(a)\}) \models q$ for q a CQ and C an $\mathcal{ALCO}$ concept, and*
- *deciding $(\mathcal{T}, \Sigma, \mathcal{A}) \models q$ for q a CQ and $\mathcal{T}, \mathcal{A}$ in $\mathcal{EL}$.*

Unfortunately, we do not have matching upper bounds. We believe that both problems are likely to be solvable in coNEXPTIME, but we are still working on a suitable algorithm.

5 Discussion and Outlook

In this paper we have given several bounds on the combined complexity of reasoning in various DLs in the presence of closed predicates, for standard reasoning problems like KB satisfiability, as well as for answering queries ranging from CQs to P2RPQs. Unlike the data-complexity, that is coNP-complete in practically all cases (from DL-Lite$_{core}$[7,19] to expressive DLs like $\mathcal{ALCHOQ}$ and $\mathcal{ALCHOI}$ [21]), the combined complexity offers a complex landscape. We summarize some results in Table 1, emphasizing the cases in which closed predicates have an interesting effect on the complexity.

Apart from establishing the precise complexity of query answering in DLs between $\mathcal{ALCO}$ and $\mathcal{ALCHOQ}$ (without closed predicates), and in all DLs between $\mathcal{EL}$ and $\mathcal{ALCHOQ}$ with closed predicates, other problems remain open. Notably, in this paper we have not considered the DL-Lite family. An algorithm for CQ entailment in DL-Lite$_\mathcal{F}$ was developed in [7] to obtain a coNP upper bound in data complexity, but it only yields very high bounds on the combined complexity that are likely not to be optimal. That algorithm deals with the intricate interactions of inverse roles, functionality, and closed predicates, that behave as nominals, and it may be possible to use the characterization of countermodels given there as a starting point for a better combined complexity upper bound. In the DL-Lite variants that do not support functionality, countermodels are likely to have a simpler structure. However, even in these simpler languages, it is not apparent whether interesting upper bounds can be obtained by simple adaptations of existing techniques. In particular, it has been shown that singleton nominals do not increase neither the data nor the combined complexity of DL-Lite [11], but the effect of the combination of nominals and (restricted) disjunction that results from closed predicates remains to be studied for the DL-Lite family.

Finally, we have seen that in general, the disjunctive information encoded by closed predicates has a negative effect on the complexity of reasoning. In the light of this, it

	Without closed predicates		With closed predicates	
	KB consistency	CQ entailment	KB consistency	CQ entailment
$\mathcal{EL}$	P [1]	NP [14,27]	ExpTime (Cor. 1.1)	$\geq$ coNExpTime (Th. 4) $\leq$ 2ExpTime (Cor. 2)
$\mathcal{ELH}^{\text{trans}}$	P [1]	$\geq$ NP,$\leq$PSpace [15]	ExpTime (Cor. 1.1)	2ExpTime (Cor. 2)
$\mathcal{ELI}$	ExpTime [2]	ExpTime [5]	ExpTime (Cor. 1.1)	2ExpTime (Cor. 2)
Horn-$\mathcal{SHOI}$ Horn-$\mathcal{SHOQ}$	ExpTime [16,22]	ExpTime [23]	ExpTime (Cor. 1.1)	2ExpTime (Cor. 2)
$\mathcal{ELIF}$ Horn-$\mathcal{SHIQ}$	ExpTime [16]	ExpTime [5]	NExpTime (Cor. 1.2)	$\geq$ N2ExpTime [10] decidable [28] / $\leq$ open*
$\mathcal{ELOIF}$, Horn-$\mathcal{SHOIQ}$	ExpTime [22]	ExpTime [23]	NExpTime (Cor. 1.2)	$\geq$ N2ExpTime [10] decidable [28] / $\leq$ open*
$\mathcal{ALCO}$	ExpTime [32,8]	$\geq$ coNExpTime (Th. 4)	ExpTime (Cor. 1.1)	$\geq$ coNExpTime (Th. 4) $\leq$ 2ExpTime (Cor. 2)
$\mathcal{SHOQ}, \mathcal{SHOI}$	ExpTime [32,12,8]	2ExpTime [4,9]	ExpTime (Cor. 1.1)	2ExpTime (Cor. 2)
$\mathcal{SHOIQ}$	NExpTime [31]	$\geq$ N2ExpTime [10] $\leq$ open*	NExpTime (Cor. 1.2)	$\geq$ N2ExpTime [10] $\leq$ open*

Table 1: Combined complexity of reasoning in description logics with/without closed predicates. By $\geq$ we indicate lower bounds, by $\leq$ upper bounds, and the rest are all completeness results. For the cells marked with *, decidability if only simple roles occur in the query follows from [28], but no complexity upper bounds are known.

seems particularly interesting to study criteria that allow to identify instances of TBoxes and queries for which the complexity does not increase. Major contributions in this direction can be found in [19]. In particular, the authors propose *safety* criteria that ensure specific TBoxes have the convexity property, guaranteeing data tractability of queries. It seems that this criteria may also be useful for establishing the existence of a *universal model* for query answering, and when a suitable representation of such a model can be built in single exponential time, query answering is likely to be feasible in single exponential time. This seems a promising direction for further investigation.

References

1. F. Baader, S. Brand, and C. Lutz. Pushing the el envelope. In *Proc. of IJCAI 2005*, pages 364–369. Morgan-Kaufmann Publishers, 2005.
2. F. Baader, S. Brandt, and C. Lutz. Pushing the el envelope further. In *Proc. of OWLED 2008*, 2008.
3. M. Cadoli, F. M. Donini, and M. Schaerf. Closed world reasoning in hybrid systems. In *Proc. of ISMIS 1990*, pages 474–481, 1990.
4. D. Calvanese, T. Eiter, and M. Ortiz. Regular path queries in expressive description logics with nominals. In *Proc. of IJCAI 2009*, pages 714–720, 2009.
5. T. Eiter, G. Gottlob, M. Ortiz, and M. Šimkus. Query answering in the description logic Horn-$\mathcal{SHIQ}$. In *Proc. of JELIA 2008*, pages 166–179, Berlin, Heidelberg, 2008. Springer-Verlag.
6. T. Eiter, C. Lutz, M. Ortiz, and M. Šimkus. Query answering in description logics with transitive roles. In *Proc. of IJCAI 2009*, volume 9, pages 759–764, 2009.
7. E. Franconi, Y. A. Ibáñez-García, and I. Seylan. Query answering with DBoxes is hard. *Electr. Notes Theor. Comput. Sci.*, 278:71–84, 2011.
8. B. Glimm, I. Horrocks, and U. Sattler. Deciding $\mathcal{SHOQ}^{\cap}$ knowledge base consistency using alternating automata. In *Proc. of DL 2008*, volume 353 of *CEUR Workshop Proceedings*, 2008.
9. B. Glimm, I. Horrocks, and U. Sattler. Unions of conjunctive queries in SHOQ. In *Proc. of KR 2008*, pages 252–262. AAAI Press/The MIT Press, 2008.
10. B. Glimm, Y. Kazakov, and C. Lutz. Status QIO: An update. In *Proc. of DL 2009*, volume 745 of *CEUR Workshop Proceedings*, 2011.
11. M. Haddad and D. Calvanese. Extending DL-Lite_A with (singleton) nominals. In *Proc. DL 2013*, volume 1014 of *CEUR Workshop Proceedings*, 2013.
12. J. Hladik. A tableau system for the description logic SHIO. In U. Sattler, editor, *IJCAR Doctoral Programme*, volume 106 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.
13. Y. Kazakov. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In *Proc. of KR 2008*, pages 274–284. AAAI Press, 2008.
14. M. Krötzsch and S. Rudolph. Conjunctive queries for $\mathcal{EL}$ with composition of roles. In *Proc. of DL 2007*, volume 250 of *CEUR Electronic Workshop Proceedings*, 2007.
15. M. Krötzsch, S. Rudolph, and P. Hitzler. Conjunctive queries for a tractable fragment of OWL 1.1. In *Proc. of ISWC 2007 + ASWC 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 310–323. Springer, 2007.
16. M. Krötzsch, S. Rudolph, and P. Hitzler. Complexities of Horn description logics. *ACM Trans. Comp. Log.*, 14(1):2:1–2:36, 2013.
17. C. Lutz. Inverse roles make conjunctive queries hard. In *Proc. of DL 2007*, pages 100–111, 2007.
18. C. Lutz. The complexity of conjunctive query answering in expressive description logics. In *Proc. of IJCAR 2008*, pages 179–193. Springer, 2008.
19. C. Lutz, I. Seylan, and F. Wolter. Ontology-based data access with closed predicates is inherently intractable(sometimes). In *Proc. of IJCAI 2013*. IJCAI/AAAI, 2013.
20. C. Lutz, I. Seylan, and F. Wolter. Ontology-mediated queries with closed predicates. In *Proc. of IJCAI 2015*. IJCAI/AAAI,, 2015.
21. M. Ortiz, D. Calvanese, and T. Eiter. Data complexity of query answering in expressive description logics via tableaux. 41(1):61–98, 2008.
22. M. Ortiz, S. Rudolph, and M. Šimkus. Worst-case optimal reasoning for the Horn-DL fragments of OWL 1 and 2. In *Proc. of KR 2010*. AAAI Press, 2010.

23. M. Ortiz, S. Rudolph, and M. Šimkus. Query answering in the Horn fragments of the description logics SHOIQ and SROIQ. In *Proc. of IJCAI 2011*, pages 1039–1044. IJCAI/AAAI, 2011.
24. M. Ortiz, M. Šimkus, and T. Eiter. Worst-case optimal conjunctive query answering for an expressive description logic without inverses. In *Proc. of AAAI 2008*, pages 504–510. AAAI Press, 2008.
25. M. Ortiz and M. Šimkus. Revisiting the hardness of query answering in expressive description logics. In *Proc. of RR 2014*, pages 216–223. Springer International Publishing, 2014.
26. R. Reiter. Equality and domain closure in first-order databases. *J. ACM*, 27(2):235–249, Apr. 1980.
27. R. Rosati. On conjunctive query answering in $\mathcal{EL}$. In *Proc. of DL 2007*, volume 250 of *CEUR Electronic Workshop Proceedings,* 2007.
28. S. Rudolph and B. Glimm. Nominals, inverses, counting, and conjunctive queries or: Why infinity is your friend! *J. of Artificial Intelligence Research*, 39:429–481, 2010.
29. K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of IJCAI 1991*, pages 466–471. Morgan Kaufmann, 1991.
30. I. Seylan, E. Franconi, and J. de Bruijn. Effective query rewriting with ontologies over DBoxes. In *Proc. of IJCAI 2009*, pages 923–925, 2009.
31. S. Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. of Artificial Intelligence Research*, 12:199–217, 2000.
32. S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.