Decidable Temporal DL-Lite with Undecidable FO-rewritability of Ontology-Mediated Queries

Extended Abstract

Alessandro Artale[1], Anton Gnatenko[1], Vladislav Ryzhikov[2] and Michael Zakharyaschev[2]

[1]*Free University of Bozen-Bolzano, Italy*
[2]*Birkbeck, University of London, U.K.*

Abstract
We design a logic in the temporal *DL-Lite* family (with non-Horn role inclusions and restricted temporalised roles), for which answering ontology-mediated atomic queries (OMAQs) can be done in ExpSpace and even in PSpace for ontologies without existential quantification in the rule heads but determining FO-rewritability or (linear) Datalog-rewritability of OMAQs is undecidable. On the other hand, we show (by reduction to monadic disjunctive Datalog) that deciding FO-rewritability of OMAQs in the non-temporal fragment of our logic can be done in 3NExpTime.

Keywords
Temporal description logics, DL-Lite, ontology-mediated query, first-order rewritability.

1. Introduction

Temporal description logics are used to reason about relational data that evolves in time. Along with the standard DL constructs on concepts and roles they admit temporal operators such as $\bigcirc_F$ (at the next moment), $\Box_F$ (always in the future), $\Diamond_F$ (sometime later) and their past-time counterparts, which give rise to *temporalised concepts* and *roles*. Being non-monodic [1], temporalised roles notoriously lead to high computational complexity even if coupled with a lightweight DL component [2, 3, 4]. Thus, any type of *DL-Lite* temporalised concept inclusions (CIs) with unguarded non-Horn temporalised role inclusions (RIs) (say, $P \sqsubseteq R \sqcup \bigcirc_F S$) result in an undecidable logic, while with Horn, Krom or core temporalised RIs the logic becomes decidable in ExpSpace or PSpace depending on the available temporal operators [3].

The proliferation of ontology-based data access [5] in the past decade has extended the list of traditional reasoning problems in DLs—such as satisfiability checking and answering ontology-mediated queries (OMQs)—with the *rewritability problem* into a target query language $\mathcal{L}$ [6, 7, 8, 9]: given an OMQ Q, decide whether there exists an $\mathcal{L}$*-rewriting* of Q, that is, an $\mathcal{L}$-query φ returning the same answers as Q over any data instance. Typical target query

DL 2023: 36th International Workshop on Description Logics, September 2–4, 2023, Rhodes, Greece
artale@inf.unibz.it (A. Artale); agnatenko@unibz.it (A. Gnatenko); vlad@dcs.bbk.ac.uk (V. Ryzhikov); michael@dcs.bbk.ac.uk (M. Zakharyaschev)
0000-0002-3852-9351 (A. Artale); 0000-0003-1499-2090 (A. Gnatenko); 0000-0002-6847-6465 (V. Ryzhikov); 0000-0002-2210-5183 (M. Zakharyaschev)

languages are the first-order logic (FO, possibly with various built-in predicates) and (linear) Datalog. For temporal OMQs given in the propositional temporal logic *LTL*, the FO-rewritability problem has recently been studied in [10]. Here, we present our initial observations on the FO- and Datalog-rewritability problems for OMQs with temporal *DL-Lite* ontologies. We design a logic with a restricted form of temporalised roles and non-Horn RIs whose syntax and potential usefulness are illustrated by the following example.

Example 1. Imagine that we are modelling the European transport network. Some passengers may like to go by plane in one direction but return by train (say, to safely bring back a selection of wines, cheeses and fine teas). To highlight such routes, one could use the following RI:

$$\mathit{flight} \sqcap \Diamond_F \mathit{train}^- \sqsubseteq \mathit{safeFlightConnection}, \tag{1}$$

where train^- denotes the inverse of the role *train* (connection) from one city to another. (In our modelling, we do not regard the roles *flight* and *train* as 'global' because they depend on the season, day of a week, etc.) We may further partition the connections into certain classes, e.g., national and international, and infer properties of connected cities based on this classification. This is achieved by means of the following non-Horn RIs:

$$\begin{aligned} &\mathit{safeFlightConnection} \sqsubseteq \mathit{national} \sqcup \mathit{international},\\ &\exists \mathit{international} \sqsubseteq \mathit{InternationalAirport}. \end{aligned} \tag{2}$$

Axiom (1) is an example a *guarded* $\Diamond$*-RI*, where the temporalised role $\Diamond_F \mathit{train}^-$ is 'guarded' by the role name *flight*. Thus, *safeFlightConnection* may be inferred at a time instant only when there exists a *flight* from one city to another and, sometime later, there is a return *train*. ⊣

The temporal description logic $\mathit{TDL\text{-}Lite}_{bool}^{[\Diamond]}$ we introduce in this paper admits arbitrary Boolean CIs with temporalised concepts, and only guarded $\Diamond$-RIs with temporalised roles that take the form

$$R_1 \sqcap \dots \sqcap R_n \sqcap \Diamond_{n+1} R_{n+1} \sqcap \dots \Diamond_k R_k \;\sqsubseteq\; R_{k+1} \sqcup \dots \sqcup R_m, \tag{3}$$

where the R_i are role names or their inverses, the $\Diamond_i$ are sequences of $\Diamond_F$ and $\Diamond_P$, and $n \geq 1$. We prove that answering ontology-mediated atomic queries (OMAQs) in $\mathit{TDL\text{-}Lite}_{bool}^{[\Diamond]}$ can be done in ExpSpace for combined complexity, and in PSpace for the *flat* fragment of $\mathit{TDL\text{-}Lite}_{bool}^{[\Diamond]}$, which disallows positive occurrences of $\exists R$ in ontology axioms (Th. 1). However, determining whether such a query can be rewritten into an FO- or a (linear) Datalog query is algorithmically undecidable (Th. 2). On the other hand, we observe that FO-rewritability of OMAQs in $\mathit{TDL\text{-}Lite}_{bool}^{[\Diamond]}$ without temporalised concepts and roles becomes decidable in 3NExpTime (Th. 3).

2. OMAQ Answering and FO-rewritability in $\mathit{TDL\text{-}Lite}_{bool}^{[\Diamond]}$

The logic $\mathit{TDL\text{-}Lite}_{bool}^{[\Diamond]}$ is a member of the *TDL-Lite* family [4, 3], which comprises temporal extensions of various languages in the atemporal *DL-Lite* family [11, 12, 13]. The alphabet of

TDL-Lite$_{bool}^{[\Diamond]}$ consists of *individual names* $a_0, a_1, \dots$, *concept names* $A_0, A_1, \dots$, and *role names* $P_0, P_1, \dots$. A *basic role* R is either a role name P_i or its inverse P_i^-. A *basic concept* C is either a concept name A_i or $\exists R$. *Temporalised concepts*, D, and *roles*, S, are defined by the grammar:

$$\begin{aligned} D &::= C \mid \bigcirc_F D \mid \bigcirc_P D \mid \Diamond_F D \mid \Diamond_P D \mid \Box_F D \mid \Box_P D, \\ S &::= R \mid \Diamond_F S \mid \Diamond_P S. \end{aligned} \tag{4}$$

A *concept inclusion* (CI) in *TDL-Lite*$_{bool}^{[\Diamond]}$ takes the form $\vartheta_1 \sqcap \dots \sqcap \vartheta_k \sqsubseteq \vartheta_{k+1} \sqcup \dots \sqcup \vartheta_{k+m}$, where the ϑ_i are temporalised concepts. A *guarded* $\Diamond$*-role inclusion* (guarded $\Diamond$-RI) takes the form (3). As usual, the empty $\sqcap$ is $\top$ and the empty $\sqcup$ is $\bot$.

A *TBox*, $\mathcal{T}$, is a finite set of CIs, and an *RBox*, $\mathcal{R}$, is a finite set of guarded $\Diamond$-RIs. Taken together, they form an *ontology*, $\mathcal{O}$, in *TDL-Lite*$_{bool}^{[\Diamond]}$. An *ABox signature*, Σ, is a set of concept and role names. A *ABox*, $\mathcal{A}$, over Σ is a finite set of facts of the form $A_i(a, \ell)$ and $P_i(a, b, \ell)$, where $A_i, P_i \in \Sigma$ and $\ell \in \mathbb{Z}$ is a *timestamp*. We denote by $ind(\mathcal{A})$ the set of individual names in $\mathcal{A}$, by $\min(\mathcal{A})$ and $\max(\mathcal{A})$ the minimal and maximal timestamps in $\mathcal{A}$, respectively, and by $tem(\mathcal{A})$ the closed interval $[\min(\mathcal{A}), \max(\mathcal{A})]$.

An *ontology-mediated atomic query* (OMAQ) is a triple $Q(x,t) = (\mathcal{O}, \Sigma, A(x,t))$, where $\mathcal{O}$ is an ontology in *TDL-Lite*$_{bool}^{[\Diamond]}$, Σ is an ABox signature, and A a concept name. (Note that the symbols in $\mathcal{O}$ and A do not have to be in Σ.) A pair $(a, \ell) \in ind(\mathcal{A}) \times tem(\mathcal{A})$ is a *certain answer* to $Q(x,t)$ over a Σ-ABox $\mathcal{A}$ if $A(a, \ell)$ is true in every model $\mathcal{M}$ of $(\mathcal{O}, \mathcal{A})$; for a detailed definition of a model the reader is referred to [3]. We denote by $ans_Q(\mathcal{A})$ the set of all certain answers to Q over $\mathcal{A}$. The *query answering problem* for Q over $\mathcal{A}$ is the decision problem for $ans_Q(\mathcal{A})$. We also consider *ontology-mediated Boolean atomic queries* (OMBAQs) of the form $Q = (\mathcal{O}, \Sigma, A)$ that require a 'yes/no' answer: a certain answer to an OMBAQ Q over $\mathcal{A}$ is 'yes' if, in every model $\mathcal{M}$ of $(\mathcal{O}, \mathcal{A})$, there exists a pair $(a, \ell) \in ind(\mathcal{A}) \times tem(\mathcal{A})$ such that $A(a, \ell)$ is true in $\mathcal{M}$, and 'no' otherwise.

With an ABox $\mathcal{A}$ we associate a temporal FO-structure $\mathfrak{S}_{\mathcal{A}}$ with domain $ind(\mathcal{A}) \times tem(\mathcal{A})$, over which we can evaluate FO($<$)- and Datalog-queries with atoms of the form $A(x,t)$, $P(x,y,t)$, $(t_1 < t_2)$. Let $\mathcal{L}$ be any relevant query language: FO($<$), linear or arbitrary Datalog queries. An OMAQ $Q(x,t) = (\mathcal{O}, \Sigma, A(x,t))$ is $\mathcal{L}$*-rewritable* if there exists an $\mathcal{L}$-query $\varphi(x,t)$, such that $ans_Q(\mathcal{A}) = \{ (a, \ell) \in ind(\mathcal{A}) \times tem(\mathcal{A}) \mid \mathfrak{S}_{\mathcal{A}} \models \varphi(a, \ell) \}$, for every Σ-ABox $\mathcal{A}$. An OMBAQ Q is $\mathcal{L}$*-rewritable* if there is an $\mathcal{L}$-query φ without answer variables such that $\mathfrak{S}_{\mathcal{A}} \models \varphi$ iff the certain answer to Q over $\mathcal{A}$ is 'yes'. It is known that FO($<$)-, linear Datalog-, and Datalog-rewritability guarantee answering OMAQs/OMBAQs in AC^0 [14], NL [15], and PTime [16] for data complexity, respectively. For the non-temporal fragment of *TDL-Lite*$_{bool}^{[\Diamond]}$, answering OMAQs/OMBAQs is ExpTime-complete for combined complexity [17]. We establsih an ExpSpace upper bound for answering OMAQs/OMBAQs in full *TDL-Lite*$_{bool}^{[\Diamond]}$. We also consider the *flat TDL-Lite*$_{bool}^{[\Diamond]}$ that disallows concepts $\exists R$ on the right-hand side of CIs. This restriction reduces the complexity of answering OMAQs/OMBAQs to PSpace.

Theorem 1. *Answering TDL-Lite*$_{bool}^{[\Diamond]}$ *OMAQs/OMBAQs is in ExpSpace and ExpTime-hard for combined complexity. Answering flat TDL-Lite*$_{bool}^{[\Diamond]}$ *OMAQs/OMBAQs is PSpace-complete for combined complexity.*

However, checking whether a query is rewritable into FO($<$) or (linear) Datalog turns out to be undecidable even for flat ontologies.

Theorem 2. FO($<$)*-rewritability, linear Datalog-rewritability* (*if NL* $\neq$ coNP), *and Datalog-rewritability* (*if* PTime $\neq$ coNP) *are undecidable for flat* $\mathit{TDL\text{-}Lite}_{bool}^{[\Diamond]}$ *OMAQs/OMBAQs.*

The proof of Th. 2 makes use of the interaction between non-Horn RIs and temporal axioms. FO-rewritability of OMAQs/OMBAQs becomes decidable in the case when ontologies do not contain temporal operators. We obtain a positive result for the language $\mathit{DL\text{-}Lite}_{bool}$ that disallows temporalised concepts and temporalised roles in $\mathit{TDL\text{-}Lite}_{bool}^{[\Diamond]}$. The proof is via a translation to monadic disjunctive Datalog, adapting a similar technique for $\mathcal{ALCI}$ [7]. However, non-Horn RIs increase the complexity from 2NExpTime to 3NExpTime.

Theorem 3. *Checking* FO*-rewritability of* $\mathit{DL\text{-}Lite}_{bool}$ *OMAQs/OMBAQs is in 3NExpTime.*

3. Related Work

The problem of deciding if a given OMQ is rewritable into a conventional query language $\mathcal{L}$ has been investigated for several important description logics. Bienvenu et al. [6] considered $\mathcal{ALC}$ and its extensions with OMAQs through the lenses of CSP and MMSNP. In particular, FO-rewritability of $\mathcal{ALCF}$ OMAQs is undecidable ($\mathcal{ALCF}$ is an extension of $\mathcal{ALC}$ with *functionality constraints* on roles). Feier et al. [7] proved 2NExpTime-completeness of FO-rewritability of OMQs with CQs in $\mathcal{ALCI}$. Lutz and Sabellek [8] established that every conjunctive OMQ in $\mathcal{EL}$ is either FO-rewritable, or linear Datalog-rewritable, or PTime-complete, and showed that each of the associated decision problems is ExpTime-complete. Gerasimova et al. [9] showed FO-rewritability to be in 2NExpTime for OMQs with CQs and the non-Horn ontology $\{A \sqsubseteq B \sqcup C\}$. Description logics with non-Horn RIs such as $\mathit{DL\text{-}Lite}_{bool}$, however, have not been considered. Separately, rewritability to first-order languages was studied by Artale et al. [18] and Kurucz et al. [10] for pure temporal logics, using automata- and group-theoretic techniques.

In this paper, we take a first step towards understanding (temporal) *DL-Lite* with guarded non-Horn role inclusions by establishing two results: a decidability procedure for FO-rewritability of OMQs in $\mathit{DL\text{-}Lite}_{bool}$, and an undecidability result for FO($<$)- and Datalog-rewritability of OMQs in $\mathit{TDL\text{-}Lite}_{bool}^{[\Diamond]}$, i.e., $\mathit{DL\text{-}Lite}_{bool}$ with temporalised concepts and (guarded) roles. An open question is if rewritability is decidable for $\mathit{DL\text{-}Lite}_{bool}$ with temporalised concepts only.

FO-rewritability (aka *boundedness*) has been studied for Datalog itself. Predicate boundedness is undecidable for binary programs [19], and even for linear programs with one binary IDB relation [20], while for linear *monadic* programs, it is in PSpace [21]. Uniform boundedness is undecidable for ternary programs [19], even if they are linear [22] (consult the latter for an explanation on different forms of Datalog boundedness). Temporalised RIs in $\mathit{TDL\text{-}Lite}_{bool}^{[\Diamond]}$ can be viewed from the Datalog perspective as using binary, or even ternary IDBs. However, the undecidability proofs of [19], [20] and [22] make use of *chains*, i.e., Datalog rules where the right-hand part contains constructs like $R(X_1, X_2) \wedge R(X_2, X_3)$, which is inexpressible in *DL-Lite*. Compared to $\mathcal{ALCF}$, *DL-Lite* lacks negation and qualified role existential predicates.

References

[1] I. M. Hodkinson, F. Wolter, M. Zakharyaschev, Monodic fragments of first-order temporal logics: 2000-2001 A.D, in: R. Nieuwenhuis, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning, 8th International Conference, LPAR 2001, Havana, Cuba, December 3-7, 2001, Proceedings, volume 2250 of *Lecture Notes in Computer Science*, Springer, 2001, pp. 1–23. URL: https://doi.org/10.1007/3-540-45653-8_1. doi:10.1007/3-540-45653-8_1.

[2] C. Lutz, F. Wolter, M. Zakharyaschev, Temporal description logics: A survey, in: S. Demri, C. S. Jensen (Eds.), 15th International Symposium on Temporal Representation and Reasoning, TIME 2008, Université du Québec à Montréal, Canada, 16-18 June 2008, IEEE Computer Society, 2008, pp. 3–14. URL: https://doi.org/10.1109/TIME.2008.14. doi:10.1109/TIME.2008.14.

[3] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, M. Zakharyaschev, First-order rewritability and complexity of two-dimensional temporal ontology-mediated queries, Journal of Artificial Intelligence Research 75 (2022) 1223–1291. URL: https://jair.org/index.php/jair/article/view/13511. doi:https://doi.org/10.1613/jair.1.13511.

[4] A. Artale, R. Kontchakov, V. Ryzhikov, M. Zakharyaschev, A cookbook for temporal conceptual data modelling with description logics, ACM Trans. Comput. Logic 15 (2014). URL: https://doi.org/10.1145/2629565. doi:10.1145/2629565.

[5] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, M. Zakharyaschev, Ontology-based data access: A survey, in: J. Lang (Ed.), Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, ijcai.org, 2018, pp. 5511–5519. URL: https://doi.org/10.24963/ijcai.2018/777. doi:10.24963/ijcai.2018/777.

[6] M. Bienvenu, B. T. Cate, C. Lutz, F. Wolter, Ontology-based data access: A study through disjunctive datalog, csp, and mmsnp, ACM Trans. Database Syst. 39 (2015). doi:10.1145/2661643.

[7] C. Feier, A. Kuusisto, C. Lutz, Rewritability in Monadic Disjunctive Datalog, MMSNP, and Expressive Description Logics, Logical Methods in Computer Science Volume 15, Issue 2 (2019). URL: https://lmcs.episciences.org/5502. doi:10.23638/LMCS-15(2:15)2019.

[8] C. Lutz, L. Sabellek, A complete classification of the complexity and rewritability of ontology-mediated queries based on the description logic el, Artificial Intelligence 308 (2022) 103709. doi:https://doi.org/10.1016/j.artint.2022.103709.

[9] O. Gerasimova, S. Kikot, A. Kurucz, V. Podolskii, M. Zakharyaschev, A tetrachotomy of ontology-mediated queries with a covering axiom, Artificial Intelligence 309 (2022) 103738. doi:https://doi.org/10.1016/j.artint.2022.103738.

[10] A. Kurucz, V. Ryzhikov, Y. Savateev, M. Zakharyaschev, Deciding fo-rewritability of regular languages and ontology-mediated queries in linear temporal logic, J. Artif. Int. Res. 76 (2023). doi:10.1613/jair.1.14061.

[11] D. Calvanese, G. De Giacomo, D. Lemho, M. Lenzerini, R. Rosati, Dl-lite: Tractable description logics for ontologies, in: Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2, AAAI'05, AAAI Press, 2005, p. 602–607.

[12] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The dl-lite family, Journal of Automated Reasoning 39 (2007) 385–429. doi:10.1007/s10817-007-9078-x.

[13] A. Artale, D. Calvanese, R. Kontchakov, M. Zakharyaschev, The DL-lite family and relations, Journal of Artificial Intelligence Research 36 (2009) 1–69. URL: https://doi.org/10.1613%2Fjair.2820. doi:10.1613/jair.2820.

[14] N. Immerman, Descriptive Complexity, Springer Verlag, 1998.

[15] E. Grädel, Capturing complexity classes by fragments of second-order logic, Theoretical Computer Science 101 (1992) 35 – 57. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0026886786&doi=10.1016%2f0304-3975%2892%2990149-A&partnerID=40&md5=80969aef1ee918e4ddcc5d494a847639. doi:10.1016/0304-3975(92)90149-A, cited by: 64.

[16] E. Dantsin, T. Eiter, G. Gottlob, A. Voronkov, Complexity and expressive power of logic programming, ACM Comput. Surv. 33 (2001) 374–425. doi:10.1145/502807.502810.

[17] R. Kontchakov, V. Ryzhikov, F. Wolter, M. Zakharyaschev, Boolean Role Inclusions in DL-Lite With and Without Time, in: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, 2020, pp. 582–591. URL: https://doi.org/10.24963/kr.2020/58. doi:10.24963/kr.2020/58.

[18] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, M. Zakharyaschev, First-order rewritability of ontology-mediated queries in linear temporal logic, Artificial Intelligence 299 (2021) 103536. URL: https://www.sciencedirect.com/science/article/pii/S0004370221000874. doi:https://doi.org/10.1016/j.artint.2021.103536.

[19] G. G. Hillebrand, P. C. Kanellakis, H. G. Mairson, M. Y. Vardi, Undecidable boundedness problems for datalog programs, The Journal of Logic Programming 25 (1995) 163–190. doi:https://doi.org/10.1016/0743-1066(95)00051-K.

[20] M. Y. Vardi, Decidability and undecidability results for boundedness of linear recursive queries, PODS '88, Association for Computing Machinery, New York, NY, USA, 1988, p. 341–351. doi:10.1145/308386.308470.

[21] R. v. d. Meyden, Predicate boundedness of linear monadic datalog is in pspace, International Journal of Foundations of Computer Science 11 (2000) 591–612. doi:10.1142/S0129054100000351.

[22] J. Marcinkowski, Achilles, turtle, and undecidable boundedness problems for small datalog programs, SIAM Journal on Computing 29 (1999) 231–257. doi:10.1137/S0097539797322140.

[23] O. Lichtenstein, A. Pneuli, Propositional temporal logics: decidability and completeness, Logic Journal of the IGPL 8 (2000) 55–85. doi:10.1093/jigpal/8.1.55.

[24] M. L. Minsky, Computation: Finite and Infinite Machines, Prentice-Hall, Inc., USA, 1967.

[25] T. Feder, M. Y. Vardi, The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory, SIAM Journal on Computing 28 (1998) 57–104. doi:10.1137/S0097539794266766.

[26] B. Larose, C. Loten, C. Tardif, A characterisation of first-order constraint satisfaction problems, in: 21st Annual IEEE Symposium on Logic in Computer Science (LICS'06), 2006, pp. 201–210. doi:10.1109/LICS.2006.6.

[27] L. Barto, The collapse of the bounded width hierarchy, Journal of Logic and Computation

26 (2014) 923–943. doi:10.1093/logcom/exu070.
[28] H. Chen, B. Larose, Asking the metaquestions in constraint tractability, ACM Trans. Comput. Theory 9 (2017). doi:10.1145/3134757.

A. Proofs

Theorem 1. *Answering TDL-Lite$_{bool}^{[\Diamond]}$ OMAQs/OMBAQs is in ExpSpace and ExpTime-hard for combined complexity. Answering flat TDL-Lite$_{bool}^{[\Diamond]}$ OMAQs/OMBAQs is PSpace-complete for combined complexity.*

Proof. We start with the formal definition of a *TDL-Lite*$_{bool}^{[\Diamond]}$ model that we need for the proof. A *TDL-Lite*$_{bool}^{[\Diamond]}$ *knowledge base* (KB) is a pair $\mathcal{K} = (\mathcal{O}, \mathcal{A})$ of an ontology and an ABox. An *interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}(n)})$, where $\Delta^{\mathcal{I}} \neq \emptyset$ and, for each $n \in \mathbb{Z}$,

$$\mathcal{I}(n) = (\Delta^{\mathcal{I}}, a_1^{\mathcal{I}}, \dots, A_1^{\mathcal{I}(n)}, \dots, P_1^{\mathcal{I}(n)}, \dots) \tag{5}$$

is a standard (non-temporal) description logic interpretation with $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, $A_i^{\mathcal{I}(n)} \subseteq \Delta^{\mathcal{I}}$ and $P_i^{\mathcal{I}(n)} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Thus, we assume that the *domain* $\Delta^{\mathcal{I}}$ and the interpretations $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ of the individual names are the same for all $n \in \mathbb{Z}$. The description logic and temporal constructs are interpreted in $\mathcal{I}(n)$ as follows:

$$(P_i^-)^{\mathcal{I}(n)} = \{ (u,v) \mid (v,u) \in P_i^{\mathcal{I}(n)} \}, \qquad (\exists S)^{\mathcal{I}(n)} = \{ u \mid (u,v) \in S^{\mathcal{I}(n)}, \text{ for some } v \},$$

$$(\Box_F D)^{\mathcal{I}(n)} = \bigcap_{k>n} D^{\mathcal{I}(k)}, \qquad (\Box_P D)^{\mathcal{I}(n)} = \bigcap_{k<n} D^{\mathcal{I}(k)}, \tag{6}$$

$$(\bigcirc_F D)^{\mathcal{I}(n)} = D^{\mathcal{I}(n+1)}, \qquad (\bigcirc_P D)^{\mathcal{I}(n)} = D^{\mathcal{I}(n-1)}, \tag{7}$$

$$(\Diamond_F S)^{\mathcal{I}(n)} = \bigcup_{k>n} S^{\mathcal{I}(k)}, \qquad (\Diamond_P S)^{\mathcal{I}(n)} = \bigcup_{k<n} S^{\mathcal{I}(k)}. \tag{8}$$

All axioms are interpreted in $\mathcal{I}$ *globally* in the sense that an inclusion $\vartheta_1 \sqcap \dots \sqcap \vartheta_k \sqsubseteq \vartheta_{k+1} \sqcup \dots \sqcup \vartheta_{k+m}$, is *true* in $\mathcal{I}$ if $\vartheta_1^{\mathcal{I}(n)} \cap \dots \cap \vartheta_k^{\mathcal{I}(n)} \subseteq \vartheta_{k+1}^{\mathcal{I}(n)} \cup \dots \cup \vartheta_{k+m}^{\mathcal{I}(n)}$, for *all* $n \in \mathbb{Z}$.

An interpretation $\mathcal{M} = (\Delta, \cdot^{\mathcal{M}})$ is called a *model* of a KB $\mathcal{K} = (\mathcal{O}, \mathcal{A})$ if all concept and role inclusions of $\mathcal{O}$ and all assertions of $\mathcal{A}$ are true in $\mathcal{M}$. A KB $\mathcal{K}$ is called consistent if it has a model.

Consistency checking for knowledge bases can be used for answering OMAQs and OMBAQs. Indeed, let $Q(x,t) = (\mathcal{O}, \Sigma, A(x,t))$ be an OMAQ. If a KB $(\mathcal{O}, \mathcal{A})$ is inconsistent, then $ans_Q(\mathcal{A}) = ind(\mathcal{A}) \times tem(\mathcal{A})$. Otherwise, $(a,k) \in ans_Q(\mathcal{A})$, if and only if $(\mathcal{O} \cup \{A \sqcap B \sqsubseteq \bot\}, \mathcal{A} \cup \{B(a,k)\})$ is inconsistent . Here $B \notin \Sigma$ is a fresh concept name. Likewise, if $Q = (\mathcal{O}, \Sigma, A)$ is an OMBAQ, then the answer for Q over $\mathcal{A}$ is "Yes" whenever $(\mathcal{O}, \mathcal{A})$ is inconsistent. If it is consistent, then the answer is "Yes", whenever $(\mathcal{O} \cup \{A \sqsubseteq \bot\}, \mathcal{A})$ is inconsistent. Therefore, it suffices to prove the respective complexity bounds for the consistency checking problem.

We first deal with the flat case. For the upper bound, we reduce flat *TDL-Lite*$_{bool}^{[\Diamond]}$ to *LTL* by grounding all concept and role inclusions of $\mathcal{O}$ with individual names from $\mathcal{A}$. For the lower bound, we note that flat *TDL-Lite*$_{bool}^{[\Diamond]}$ contains *LTL* as a sub-language. Then the theorem follows from the fact that *LTL* is PSpace-complete [23].

For the general case of *TDL-Lite*$_{bool}^{[\Diamond]}$, the ExpTime-hardness follows from the fact that it extends *DL-Lite*$_{g\text{-}bool}$, which is ExpTime-complete [17].

We now prove the ExpSpace upper bound. Fix a *TDL-Lite*$_{bool}^{[\Diamond]}$ knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{A})$. As a first step, we translate $\mathcal{K}$ to a theory $\mathfrak{T}_{\mathcal{K}}$ in the first-order temporal logic $\mathcal{QTL}$. On the second step we provide a translation of $\mathfrak{T}_K$ to pure *LTL*. The size of the resulting theory will be exponential in the size of $\mathcal{K}$. Since *LTL* is PSpace-complete, this yields the ExpSpace algorithm for *TDL-Lite*$_{bool}^{[\Diamond]}$.

We define the first-order temporal logic $\mathcal{QTL}$ following Artale et al. [4], but restricting to operators $\bigcirc, \Diamond, \Box$. Its language consists of variables $\{x_1, x_2, \dots\}$, constants $\{a_1, a_2, \dots\}$, and predicates $\{A_1, A_2, \dots\}$ with associated arities. A predicate of arity zero is called a *proposition*. Formulas φ of $\mathcal{QTL}$ are defined by the following grammar:

$$\begin{array}{lcl} \varphi & ::= & A_i(t_1, \dots, t_{k_i}) \mid \bot \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x \varphi \mid O\varphi \\ O & ::= & \bigcirc_F \mid \bigcirc_P \mid \Diamond_F \mid \Diamond_P \mid \Box_F \mid \Box_P \end{array}$$

Here, k_i is the arity of A_i, t_j are *terms*, i.e. variables or constants, and x is a variable.

An interpretation $\mathcal{I} = \{\Delta, \cdot^{\mathcal{I}}\}$ gives, for every $n \in \mathbb{Z}$, a first-order structure

$$\mathcal{I}(n) = (\Delta, a_1^{\mathcal{I}}, \dots, A_1^{\mathcal{I}(n)}, \dots).$$

As in case of *TDL-Lite*$_{bool}^{[\Diamond]}$ interpretations, we assume that the domain Δ and the interpretations of the constants $a_i^{\mathcal{I}} \in \Delta$ are the same for all n, while the interpretations of predicates $A_i^{\mathcal{I}(n)} \subseteq \Delta^{k_i}$ may differ. An *assignment* σ maps variables to the elements of Δ. The semantics of $\mathcal{QTL}$ are defined in terms of an interpretation and an assignment. For a term t the symbol $t^{\mathcal{I},\sigma}$ denotes $t^{\mathcal{I}}$ if t is a constant and $\sigma(t)$ if it is a variable.

$$\begin{array}{l}
\mathcal{I}, n, \sigma \models A_i(t_1, \dots, t_{k_i}) \iff (t_1^{\mathcal{I},\sigma}, \dots, t_{k_i}^{\mathcal{I},\sigma}) \in A_i^{\mathcal{I}(n)}; \\
\mathcal{I}, n, \sigma \not\models \bot; \\
\mathcal{I}, n, \sigma \models \neg\varphi \iff \mathcal{I}, n, \sigma \not\models \varphi; \\
\mathcal{I}, n, \sigma \models \varphi \wedge \psi \iff \mathcal{I}, n, \sigma \models \varphi \text{ and } \mathcal{I}, n, \sigma \models \psi; \\
\mathcal{I}, n, \sigma \models \forall x \varphi \iff \mathcal{I}, n, \sigma' \models \varphi \text{ for all } \sigma' \text{ that differ from } \sigma \text{ on } x \text{ only}; \\
\mathcal{I}, n, \sigma \models \bigcirc_F \varphi \iff \mathcal{I}, n+1, \sigma \models \varphi; \\
\mathcal{I}, n, \sigma \models \bigcirc_P \varphi \iff \mathcal{I}, n-1, \sigma \models \varphi; \\
\mathcal{I}, n, \sigma \models \Diamond_F \varphi \iff \mathcal{I}, m, \sigma \models \varphi \text{ for some } m > n; \\
\mathcal{I}, n, \sigma \models \Diamond_P \varphi \iff \mathcal{I}, m, \sigma \models \varphi \text{ for some } m < n; \\
\mathcal{I}, n, \sigma \models \Box_F \varphi \iff \mathcal{I}, m, \sigma \models \varphi \text{ for all } m > n; \\
\mathcal{I}, n, \sigma \models \Box_P \varphi \iff \mathcal{I}, m, \sigma \models \varphi \text{ for all } m < n.
\end{array}$$

We use standard abbreviations: $\top = \neg\bot$, $\varphi \vee \psi = \neg(\neg\varphi \wedge \neq \psi)$, and $\exists x \varphi = \neg\forall x \neg\varphi$. If a formula φ contains no free variables (i.e. is a *sentence*), then we omit the assignment σ in $\mathcal{I}, n, \sigma \models \varphi$ and write $\mathcal{I}, n \models \varphi$.

An interpretation $\mathcal{I}$ is a model for a set of $\mathcal{QTL}$ sentences (a *theory*) $\mathfrak{T}$, if $\mathcal{I}, 0 \models \varphi$ for all $\varphi \in \mathfrak{T}$. A theory is called consistent if it has a model.

We would like to obtain a finite $\mathcal{QTL}$ theory $\mathfrak{T}_{\mathcal{K}}$, such that it is consistent if and only if $\mathcal{K}$ is so. The translation follows that of Artale et al. [4] with some modifications to treat the non-Horn role inclusions of *TDL-Lite*$^{[\Diamond]}_{bool}$.

Recall that we have basic concepts and roles, and temporalized concepts and roles, defined by (4). A basic concept is either a concept name A_i or a symbol of the form $\exists R$, where R is a basic role. A basic role is either a role name P_i or its inverse P_i^-.

Given an ontology $\mathcal{O} = \mathcal{T} \cup \mathcal{R}$, a set of basic roles $\mathbf{R} = \{R_1, \dots, R_k\}$ is called a *role type*, if the following two conditions hold:

1. The interpretation $\mathcal{I}_{\mathbf{R}} = \big(\{a, b\}, \cdot^{\mathcal{I}}\big)$, such that $R^{\mathcal{I}} = \{(a, b)\}$ for all $R \in \mathbf{R}$ and $R^{\mathcal{I}} = \varnothing$ otherwise, is a model of $\mathcal{R}$.
2. The knowledge base $\mathcal{K}_{\mathbf{R}} = \big(\mathcal{T}, \{\exists R(a), \exists R^-(b) \mid R \in \mathbf{R}\}\big)$ is consistent.

The signature of $\mathfrak{T}_{\mathcal{K}}$ includes the following symbols:

- For each concept name A_i, a unary predicate $A_i(x)$;
- For each basic role R, a binary predicate $R(x, y)$, two unary $ER(x), ER^-(x)$. If $\bot$ is used in role inclusions of $\mathcal{K}$, we treat $E\bot, E\bot^-$ simply as $\bot$;
- For each $a \in ind(\mathcal{A})$, a constant symbol a.

Naturally, for a temporalized concept $D = op^* A$, where A is a concept name and op^* is a sequence of temporal operators, we can write a $\mathcal{QTL}$ formula $D(x) = op^* A(x)$. Further, for a temporalized concept $D = op^* \exists R$ we write $D(x) = op^* ER(x)$ in $\mathcal{QTL}$. Similarly, for a temporalized role $S = op^* R$ we write $S(x, y) = op^* R(x, y)$.

We now are ready to define $\mathfrak{T}_{\mathcal{K}}$ as the minimal set of formulas satisfying the following conditions. For each concept inclusion, $D_1 \sqcap \dots \sqcap D_k \sqsubseteq D_{k+1} \sqcup \dots \sqcup D_{k+m}$, $\mathfrak{T}_{\mathcal{K}}$ contains

$$\Box_F \Box_P \forall x \, \big(D_1(x) \wedge \dots \wedge D_k(x) \longrightarrow D_{k+1}(x) \vee \dots \vee D_{k+m}(x)\big). \tag{9}$$

Similarly, for each role inclusion, $S_1 \sqcap \dots \sqcap S_k \sqsubseteq S_{k+1} \sqcup \dots \sqcup S_{k+m}$, $\mathfrak{T}_{\mathcal{K}}$ contains

$$\Box_F \Box_P \forall x \forall y \, \big(S_1(x, y) \wedge \dots \wedge S_k(x, y) \longrightarrow S_{k+1}(x, y) \vee \dots \vee S_{k+m}(x, y)\big). \tag{10}$$

To encode the semantics of roles, $\mathfrak{T}_{\mathcal{K}}$ contains

$$\Box_F \Box_P \forall x \forall y \, R(x, y) \longrightarrow ER(x) \wedge ER(y) \tag{11}$$

$$\Box_F \Box_P \forall x \forall y \, R(x, y) \longrightarrow R^-(x, y). \tag{12}$$

To encode the behavior of roles in the anonymous part, we use a projection of role types to pairs of monadic concepts. Let $\mathrm{Tp}(R)$ denote the collection of all role types that contain the basic role R. For $\mathbf{R} = \{R_1, \dots, R_k\}$, let $\mathcal{E}_{\mathbf{R}}(x)$ denote the conjunction

$$ER_1(x) \wedge \dots \wedge ER_k(x)$$

and $\mathcal{E}^-_{\mathbf{R}}(x)$ denote the conjunction

$$ER_1^-(x) \wedge \dots \wedge ER_k^-(x).$$

Then, for each basic role R in $\mathcal{K}$, $\mathfrak{T}_{\mathcal{K}}$ contains the following axiom:

$$\Box_F \Box_P \forall x \left(ER(x) \quad \longrightarrow \bigvee_{\mathbf{R} \in \mathrm{Tp}(R)} \left(\mathcal{E}_{\mathbf{R}}(x) \wedge \exists y\, \mathcal{E}^{-}_{\mathbf{R}}(y)\right)\right) \tag{13}$$

Finally, $\mathfrak{T}_{\mathcal{K}}$ contains the following ABox representation:

$$\bigwedge_{A(a,k) \in \mathcal{A}} \bigcirc^k A(a) \quad \wedge \bigwedge_{P(a,b,k) \in \mathcal{A}} \bigcirc^k P(a,b). \tag{14}$$

This finishes the construction of $\mathfrak{T}_{\mathcal{K}}$.

Lemma 1. *The theory $\mathfrak{T}_{\mathcal{K}}$ is consistent if and only if $\mathcal{K}$ is consistent. Moreover, $|\mathfrak{T}_{\mathcal{K}}| = 2^{\mathrm{poly}(|\mathcal{K}|)}$.*

Proof. It can be verified from the description above that $|\mathfrak{T}_{\mathcal{K}}| = 2^{\mathrm{poly}(|\mathcal{K}|)}$. We now show that the consistency of $\mathcal{K}$ coincides with the consistency of $\mathfrak{T}_{\mathcal{K}}$.

($\Rightarrow$) Let $\mathcal{M} = \left(\Delta, \cdot^{\mathcal{M}}\right)$ be a model of $\mathcal{K}$. We construct a model $\mathcal{N} = \left(\Delta, \cdot^{\mathcal{N}}\right)$ for $\mathfrak{T}_{\mathcal{K}}$ with the same domain Δ. For every $a \in \mathit{ind}(\mathcal{A})$ we set $a^{\mathcal{N}} = a^{\mathcal{M}}$. For each concept name A we set $A^{\mathcal{N}} = A^{\mathcal{M}}$ and for each basic role R we set $R^{\mathcal{N}} = R^{\mathcal{M}}$ and $ER^{\mathcal{N}} = (\exists R)^{\mathcal{M}}$. By construction, $\mathcal{N}$ satisfies the (9)—(14), so it is a model of $\mathfrak{T}_{\mathcal{K}}$.

($\Leftarrow$) Conversely, suppose $\mathcal{N} = \left(\Delta, \cdot^{\mathcal{N}}\right)$ is a model of $\mathfrak{T}_{\mathcal{K}}$. We construct a model $\mathcal{M} = \left(\Delta^{\omega}, \cdot^{\mathcal{M}}\right)$ for $\mathcal{K}$ with the domain $\Delta^{\omega} = \bigcup_{n \in \mathbb{N}} \Delta \times \{n\}$. In other words, for each natural number n we create a copy of Δ, and the domain of $\mathcal{M}$ is a disjoint union of these copies. By $\langle u, n\rangle$ we denote the copy of $u \in \Delta$ that belongs to the nth copy of Δ in Δ^{ω}.

We now define the interpretation function. For each $a \in \mathit{ind}(\mathcal{A})$ we set $a^{\mathcal{M}} = \langle a^{\mathcal{N}}, 0\rangle$. Further, for each concept name A, $u \in \Delta$, $n \in \mathbb{N}$, and $k \in \mathbb{Z}$, we set $\langle u, n\rangle^{\mathcal{M}} \in A^{\mathcal{M}(k)}$ if $a^{\mathcal{N}} \in A^{\mathcal{N}(k)}$. The interpretation of roles is done in two steps. Fix a $k \in \mathbb{Z}$. First, for each basic role R and each $u, v \in \Delta$, and $n \in \mathbb{N}$, we let $\left(\langle u, n\rangle, \langle v, n\rangle\right) \in R^{\mathcal{M}(k)}$ if $(u, v) \in R^{\mathcal{N}(k)}$. Second, for each basic role R, consider any $u \in ER^{\mathcal{N}(k)}$. By (13) there exist a role type $\mathbf{R}$ and $y \in \Delta$, such that $R \in \mathbf{R}$, and $\mathcal{N}, k \models \mathcal{E}_{\mathbf{R}}(u)$ and $\mathcal{N}, k \models \mathcal{E}^{-}_{\mathbf{R}}(y)$. Then for each $W \in \mathbf{R}$ and $n \in \mathbb{N}$ we set $\left(\langle u, n\rangle, \langle y, n+1\rangle\right) \in W^{\mathcal{M}(k)}$. This finishes the construction of $\mathcal{M}$.

It remains to verify that $\mathcal{M}$ is a model of $\mathcal{K}$. Observe that, by construction, for each concept name A, $A^{\mathcal{M}} = A^{\mathcal{N}}$. Moreover, for each basic role R, $(\exists R)^{\mathcal{M}} = ER^{\mathcal{N}}$. Since $\mathcal{N}$ satisfies (9), $\mathcal{M}$ satisfies $\mathcal{T}$. Then, consider any $\langle u, n\rangle, \langle v, m\rangle \in \Delta^{\omega}$. If $n = m$, then the roles between the two elements were introduced on the first step and taken from $\mathcal{N}$. By (10), they form a role type. If $n \neq m$, then the roles between the two elements were introduced on the second step and by construction form a role type. It follows that $\mathcal{M}$ satisfies $\mathcal{R}$. $\square$

The next step is to further translate $\mathfrak{T}_{\mathcal{K}}$ to a pure *LTL* theory. First, we shall get rid of the existential quantifier in the formula (13). Namely, for each role type $\mathbf{R}$ we add to the signature a proposition $B_{\mathbf{R}}$ and a constant $d_{\mathbf{R}^-}$. Then, let $\mathfrak{T}'_{\mathcal{K}}$ be a $\mathcal{QTL}$ theory obtained from $\mathfrak{T}_{\mathcal{K}}$ by substituting all formulas of type (13) with

$$\Box_F \Box_P \forall x \left(ER(x) \quad \longrightarrow \bigvee_{\mathbf{R} \in \mathrm{Tp}(R)} \mathcal{E}_{\mathbf{R}}(x) \wedge B_{\mathbf{R}} \right) \tag{15}$$

$$B_{\mathbf{R}} \longrightarrow \mathcal{E}^{-}_{\mathbf{R}}(d_{\mathbf{R}^-}) \tag{16}$$

Note that the formula (16) talks about the moment of time 0. Moreover, it is still the case that $|\mathfrak{T}'_{\mathcal{K}}| = 2^{\text{poly}}(|\mathcal{K}|)$.

Lemma 2. *The theory $\mathfrak{T}'_{\mathcal{K}}$ is consistent if and only if $\mathfrak{T}_{\mathcal{K}}$ is consistent.*

Proof.
($\Rightarrow$) Let $\mathcal{N} = (\Delta', \cdot^{\mathcal{N}})$ be a model of $\mathfrak{T}'_{\mathcal{K}}$. We construct a model $\mathcal{M} = (\Delta, \cdot^{\mathcal{M}})$ of $\mathfrak{T}_{\mathcal{K}}$ as follows. Let $\mathcal{N}[k] = (\Delta'_k, \cdot^{\mathcal{N}[k]})$ denote a $\mathcal{QTL}$ interpretation with $\Delta'_k = \Delta' \times \{k\}$ and $\cdot^{\mathcal{N}[k](n)} = \cdot^{\mathcal{N}(n+k)}$. Further, let $\Delta = \Delta' \cup \bigcup_{k \in \mathbb{Z}} \Delta'_k$. Define $\cdot^{\mathcal{M}}$ so that on $ind(\mathcal{A})$ it coincides with $\cdot^{\mathcal{N}}$, and for predicates it coincides with $\cdot^{\mathcal{N}}$ on Δ' and with $\cdot^{\mathcal{N}[k]}$ on Δ'_k. Clearly, $\mathcal{M}$ is a model of $\mathfrak{T}'_{\mathcal{K}}$. To see that $\mathcal{M}$ is also a model of $\mathfrak{T}_{\mathcal{K}}$, observe that it satisfies the formula (13), since for each $\mathbf{R}$ and each $k \in \mathbb{Z}$ it has an element $\langle d_{\mathbf{R}^-}, k \rangle$, such that

$$\Box_F \Box_P \forall x \left(ER(x) \quad \longrightarrow \bigvee_{\mathbf{R} \in \mathrm{Tp}(R)} \left(\mathcal{E}_{\mathbf{R}}(x) \wedge \mathcal{E}^{-}_{\mathbf{R}}(\langle d_{\mathbf{R}^-}, k \rangle) \right) \right) \tag{17}$$

is true in $\mathcal{M}$.
($\Leftarrow$) Let $\mathcal{M} = (\Delta, \cdot^{\mathcal{M}})$ be a model of $\mathfrak{T}_{\mathcal{K}}$. We create a model $\mathcal{N} = (\Delta', \cdot^{\mathcal{N}})$ for $\mathfrak{T}'_{\mathcal{K}}$. For each role type $\mathbf{R}$ consider the following interpretation $\mathcal{M}_{\mathbf{R}} = (\Delta_{\mathbf{R}}, \cdot^{\mathcal{M}_{\mathbf{R}}})$. There are two possible cases. Case A: there exists $k \in \mathbb{Z}$ and $\delta_{\mathbf{R}} \in \Delta$, with $\mathcal{M}, k \models \mathcal{E}^{-}_{\mathbf{R}}(\delta_{\mathbf{R}})$. Then let $\mathcal{M}_{\mathbf{R}} = \mathcal{M}[-k]$. Case B: there is no such k and u. Then let $\mathcal{M}_{\mathbf{R}} = \mathcal{M}$.

We set $\Delta' = \Delta \cup \bigcup_{\mathbf{R}} \Delta_{\mathbf{R}}$ and define $\cdot^{\mathcal{N}}$ as follows. On $ind(\mathcal{A})$ it coincides with $\cdot^{\mathcal{M}}$. For each role type $\mathbf{R}$, in Case A we set $B^{\mathcal{N}}_{\mathbf{R}}$ to be true, and $d^{\mathcal{N}}_{\mathbf{R}^-} = \delta_{\mathbf{R}}$. In Case B we set $B^{\mathcal{N}}_{\mathbf{R}}$ to be false, and $d^{\mathcal{N}}_{\mathbf{R}^-}$ to be any $\delta \in \Delta'$. For predicates, $\cdot^{\mathcal{N}}$ coincides with $\cdot^{\mathcal{M}}$ on Δ and $\cdot^{\mathcal{M}_{\mathbf{R}}}$ on $\Delta_{\mathbf{R}}$. Then $\mathcal{N}$ satisfies (15) and (16), hence it is a model of $\mathfrak{T}'_{\mathcal{K}}$. □

The theory $\mathfrak{T}'_{\mathcal{K}}$ belongs to the universal fragment of $\mathcal{QTL}$. Namely, all the variables in its formulas are universally quantified. This allows us to *ground* $\mathfrak{T}'_{\mathcal{K}}$ by substituting variables in its formulas in all possible ways with constants from $ind(\mathcal{A}) \cup \{d_{\mathbf{R}^-} \mid \mathbf{R} \text{ is a role type}\}$. Since the predicates in $\mathfrak{T}'_{\mathcal{K}}$ are of arity at most 2, that gives an *LTL* theory $\mathfrak{T}''_{\mathcal{K}}$ with size polynomial in the size of $\mathfrak{T}'_{\mathcal{K}}$. This allows us to conclude the proof of the theorem with the following lemma.

Lemma 3. *There exists an LTL theory $\mathfrak{T}''_{\mathcal{K}}$ that is consistent if and only if $\mathcal{K}$ is consistent. Moreover, $|\mathfrak{T}''_{\mathcal{K}}| = 2^{\text{poly}(|\mathcal{K}|)}$.*

□

Theorem 2. FO$(<)$*-rewritability, linear Datalog-rewritability (if NLogSpace $\neq$ coNP), and Datalog-rewritability (if PTime $\neq$ coNP) are undecidable for flat TDL-Lite$^{[\Diamond]}_{bool}$ OMAQs (OMBAQs).*

Proof. We reduce the halting problem of 2-counter machines to deciding FO($<$)- and (linear) Datalog-rewritability of OMAQs/OMBAQs. Recall, that a *2-counter machine* is defined by a finite set of states S with distinguished *initial state* s^{init} and *final state* s^{fin}, two counters able to store non-negative integers, and a transition function Θ. On each step the machine performs a *transition* by changing its state and incrementing or decrementing each counter by 1 with a restriction that their values remain non-negative. The next transition is chosen according to the current state and the values of the counters. However, the machine is only allowed to perform zero-tests on counters and does not distinguish between two different positive values.

Formally, transitions are given by a partial function

$$\Theta\colon S \times \{0, +\} \times \{0, +\} \to S \times \{-1, 0, 1\} \times \{-1, 0, 1\},$$

defined for each triple (s, c, d) except for the case $s = s^{fin}$.

Let $\mathrm{sgn}(0) = 0$ and $\mathrm{sgn}(k) = +$ for all $k > 0$. A *computation* of M is a sequence of *configurations*, that is, of triples

$$(s_0, c_0, d_0), (s_1, c_1, d_1), (s_2, c_2, d_2) \dots (s_n, c_n, d_n),$$

such that for each $i \geq 0$ holds $\Theta(s_i, \mathrm{sgn}(c_i), \mathrm{sgn}(d_i)) = (s_{i+1}, \varepsilon_1, \varepsilon_2)$ and $c_{i+1} = c_i + \varepsilon_1$, $d_{i+1} = d_i + \varepsilon_2$. We assume that $c_0, d_0 \geqslant 0$ and Θ is such that $c_i, d_i \geqslant 0$ for all i. A *proper* computation is the (unique) one with $s_0 = s^{init}$, $c_0, d_0 = 0$ and $s_n = s^{fin}$.

We say that M *halts* if it has a proper computation, that is, if starting from $(s^{init}, 0, 0)$ it reaches s^{fin} sooner or later. It was shown by Minsky [24] that 2-counter machines can simulate Turing machines and therefore the halting problem for 2-counter machines is undecidable.

Let M be a 2-counter machine. We construct a OMBAQ $Q_M = (\mathcal{O}, \Sigma, \mathit{Yellow})$, answering which is in AC^0 if M does not halt, but becomes coNP-hard if M halts. The latter is achieved by reduction of graph (non) 3-colorability problem. Answering OMAQs/OMBAQs, rewritable to FO($<$), linear Datalog and Datalog, is, respectively, in AC^0, NLogSpace, and PTime, for data complexity. Therefore, modulo the respective assumptions on complexity classes, deciding the rewritability of Q_M into each of these languages would allow to solve the halting problem for M. In the end of the section, we generalize this construction to OMAQs.

The ABox signature Σ contains a concept name I and role names R, V_1, V_2, E. We first describe the ontology $\mathcal{O}$. Apart from the symbols of Σ, we will use other concept and role names that do not appear in ABoxes, but can be inferred using the concept and role inclusions of $\mathcal{O}$. From now on, we assume that all ABoxes are over Σ.

To simulate the computations of M we need a representation of its states, counters and transitions. That is, for each state $s_i \in S$ and every $\alpha, \beta \in \{0, +\}$, representing the signs of the two counters, we create a concept name $S_i^{\alpha\beta}$ and restrict such concepts to be global.

$$S_i^{\alpha\beta} \sqcap S_j^{\gamma\delta} \sqsubseteq \bot, \qquad \textit{for all } (i, \alpha, \beta) \neq (j, \gamma, \delta), \tag{18}$$

$$S_i^{\alpha\beta} \sqsubseteq \Box_F \Box_P S_i^{\alpha\beta}, \qquad \textit{for all } (i, \alpha, \beta). \tag{19}$$

For brevity, assume that $s^{init} = s_0$ and $s^{fin} = s_f$.

To encode a machine configuration we use a "zero" concept name Z, and two concept names C_1 and C_2 representing the two counters. These concepts are forced to be “instantaneous”:

$$Z \sqcap \Diamond_F Z \sqsubseteq \bot \qquad C_1 \sqcap \Diamond_F C_1 \sqsubseteq \bot \qquad C_2 \sqcap \Diamond_F C_2 \sqsubseteq \bot \tag{20}$$

If a counter is different from zero then it must hold at a time point different from that one where Z holds, called in the following *zero point.* This is captured by the following axiom:

$$Z \sqcap C_1^{\alpha} \sqcap C_2^{\beta} \sqcap S_i^{\gamma\delta} \sqsubseteq \bot, \tag{21}$$

for all $(\alpha, \beta) \neq (\gamma, \delta)$, with $\alpha, \beta, \gamma, \delta \in \{0, +\}$, and all $s_i \in S$. Here $C_j^0 = C_j$ and $C_j^+ = \Diamond_F C_j$, for $j = 1, 2$.

The transitions between configurations are regulated by axioms on roles. For state transitions we use the role name $R \in \Sigma$ and make sure that such a "state transition" is done at a zero point:

$$\exists R \sqsubseteq Z, \qquad \exists R^- \sqsubseteq Z. \tag{22}$$

Next, we claim that the states connected by R comply with Θ. For each $\xi = (s_i, \alpha, \beta) \in S \times \{0, +\}^2$ and each $s_j \in S$, such that $\Theta(s_i, \alpha, \beta) = (s_j, \varepsilon_1, \varepsilon_2)$, for some $\varepsilon_1, \varepsilon_2 \in \{-1, 0, 1\}$, we add a role name $R_{s_j}^{\xi}$ together with the following axioms:

$$R \sqsubseteq \bigsqcup_{\xi, s_j} R_{s_j}^{\xi}, \qquad \exists R_{s_j}^{\xi} \sqsubseteq S_i^{\alpha\beta}, \qquad \exists \left(R_{s_j}^{\xi}\right)^- \sqsubseteq \bigsqcup_{\alpha', \beta'} S_j^{\alpha'\beta'}. \tag{23}$$

It remains to ensure that the transitions to the state encoded by $S_j^{\alpha'\beta'}$ respect the values of the counters in the state s_j. For each counter C_j, $j = 1, 2$, we use the role name $V_j \in \Sigma$ and, for all $\xi = (s_i, \alpha, \beta)$, the following role axioms.

$$\exists V_j \sqsubseteq C_j, \qquad V_j \sqsubseteq \bigsqcup_{\xi} V_j^{\xi}, \qquad \exists V_j^{\xi} \sqsubseteq S_i^{\alpha\beta}, \qquad \exists \left(V_j^{\xi}\right)^- \sqsubseteq \bigcirc_j^{\xi} C_j, \tag{24}$$

where, $\bigcirc_j^{\xi}$ is $\bigcirc_P$, $\bigcirc_F$, or the empty string, depending on whether $\Theta(\xi) = (s, \varepsilon_1, \varepsilon_2)$ with ε_j being -1, 1, or 0, respectively.

We also use the concept name $I \in \Sigma$ as a marker of a start of the computation:

$$I \sqsubseteq S_0^{00}. \tag{25}$$

Let $\mathcal{O}_{machine}$ be the set of axioms (18)-(25). It can be used to verify what we call "computation paths", that is, whether there exists a sequence of individuals of an ABox, $\mathcal{A}$, representing a configuration of a 2-counter machine, M.

Formally, given an ABox $\mathcal{A}$ over Σ, such that $(\mathcal{O}_{machine}, \mathcal{A})$ is consistent, we say that $\mathcal{A}$ has a *computation path* if there is a pair (σ, m) with $m \in \mathbb{Z}$ and σ being a sequence of individuals $a_0, a_1, \ldots, a_n$ of $\mathcal{A}$ such that the following properties hold:

- $R(a_i, a_{i+1}, m) \in \mathcal{A},\ 0 \leqslant i < n$;
- there is $m_1^i \geqslant m$ such that $V_1(a_i, a_{i+1}, m_1^i) \in \mathcal{A},\ 0 \leqslant i < n$;
- there is $m_2^i \geqslant m$ such that $V_2(a_i, a_{i+1}, m_2^i) \in \mathcal{A},\ 0 \leqslant i < n$.

We call n the *length* of the computation path. With *proper computation path* we denote a computation path such that $I(a_0, m) \in \mathcal{A}$ and $(\mathcal{O}, \mathcal{A}) \models S_f^{\alpha\beta}(a_n, m)$, for some $\alpha, \beta \in \{0, +\}$.

We need some means to distinguish proper computations from non-proper ones. The idea is to check for the "parallel" roles R, V_1, V_2, find the beginning of the computation path marked with I, and traverse the computation path till the end to check for the final state. To this end, we introduce role names $T, T^{00}, T^{01}, T^{11}$ and concept names $A, \overline{A}, B, \overline{B}$ together with new axioms. Role T encodes the notion of "transition" and connects subsequent individuals of the computation path. We start by enforcing that every correct transition results in T with the following axioms:

$$R \sqcap \triangle V_1 \sqcap \triangle V_2 \sqsubseteq T, \tag{26}$$

where $\triangle$ is either the empty string or the temporal operator $\Diamond_F$ giving thus four different axioms.

Furthermore, the concept name A should hold in the beginning of the configuration path, and then should spread along the subsequent individuals of the computation path. Therefore, A holds in the initial state of the computation path:

$$I \sqsubseteq A, \tag{27}$$

and should remain true only along proper computation paths:

$$A \sqcap \overline{A} \sqsubseteq \bot, \qquad T \sqsubseteq T^{00} \sqcup T^{01} \sqcup T^{11}, \qquad \exists T^{ab} \sqsubseteq A^a, \qquad \exists \left(T^{ab}\right)^{-} \sqsubseteq A^b, \tag{28}$$

with $A^0 = \overline{A}$ and $A^1 = A$. Intuitively, T^{11} labels proper computation paths while both T^{00} and T^{01} are labeling non-proper ones. If A holds at the end of a computation path, then, the path is a proper computation path and we signal this by making true also B:

$$A \sqcap \bigsqcup_{\alpha\beta} S_f^{\alpha\beta} \sqsubseteq B \tag{29}$$

Let $\mathcal{O}_{link}$ comprise axioms (26)-(29). This ontology 'links' the computations of M, axiomatised by $\mathcal{O}_{machine}$, to our original plan of reducing graph (non) 3-colorability to answering Q_M. Assume, that an ABox contains a proper computation path and a graph, constituted by edges of the role name E. The idea is to use B as a trigger to initialize the coloring of this graph.

If B was inferred somewhere at a time point m, it spreads over instances of the role E, that represent edges of the graph. These is achieved in the same fashion as before via the following axioms:

$$B \sqcap \overline{B} \sqsubseteq \bot, \qquad E \sqsubseteq E^{11} \sqcup E^{00} \tag{30}$$

$$\exists E^{00} \sqsubseteq \overline{B}, \qquad \exists \left(E^{00}\right)^{-} \sqsubseteq \overline{B}, \tag{31}$$

$$\exists E^{11} \sqsubseteq B, \qquad \exists \left(E^{11}\right)^{-} \sqsubseteq B. \tag{32}$$

Finally, we require all B-labeled to be properly colored in four colors, *Red, Blue, Green, Yellow.* That is:

$$B \sqsubseteq \textit{Red} \sqcup \textit{Blue} \sqcup \textit{Green} \sqcup \textit{Yellow}. \tag{33}$$

We make colors pairwise disjoint. Let $\mathcal{C}_1, \mathcal{C}_2$ range over the colors. Then:

$$\mathcal{C}_1 \sqcap \mathcal{C}_2 \sqsubseteq \bot, \text{ for all } \mathcal{C}_1 \neq \mathcal{C}_2. \tag{34}$$

Further, the coloring is correct, that is, E always connects two distinct colors. This is obtained introducing role names $E_{\mathcal{C}_1,\mathcal{C}_2}$, for each $(\mathcal{C}_1,\mathcal{C}_2), \mathcal{C}_1 \neq \mathcal{C}_2$, and the axioms:

$$E \sqsubseteq \bigsqcup_{\mathcal{C}_1 \neq \mathcal{C}_2} E_{\mathcal{C}_1,\mathcal{C}_2}, \qquad \exists E_{\mathcal{C}_1,\mathcal{C}_2} \sqsubseteq \mathcal{C}_1, \qquad \exists E^{-}_{\mathcal{C}_1,\mathcal{C}_2} \sqsubseteq \mathcal{C}_2. \tag{35}$$

Remark 1. *These axioms will only trigger the coloring of the E-connected component attached to the end of the computation path. However, since 3-colorability is NP-hard already for connected graphs, this is not a problem.*

We set the ontology $\mathcal{O}_{color}$ to contain the axioms (30)-(35), and $\mathcal{O} = \mathcal{O}_{machine} \cup \mathcal{O}_{link} \cup \mathcal{O}_{color}$, and let $Q_M = (\mathcal{O}_M, \Sigma, \mathit{Yellow})$.

Intuitively, if $\mathcal{A}$ is an ABox over Σ, such that $(\mathcal{O},\mathcal{A})$ is consistent, and M does not halt, then there is always a model of $(\mathcal{O},\mathcal{A})$ with no *Yellow*. Therefore, the answer to Q_M is always 'no'. If, however, M halts, then Q_M checks for (non) 3-colorability of an E-graph contained in $\mathcal{A}$ at the same time point as the proper computation path.

We now formalize this intuition. We assume that all ABoxes are formulated in terms of Σ.

Lemma 4. *There exists an* FO$(<)$*-formula φ_M such that for an arbitrary ABox $\mathcal{A}$ it holds that $(\mathcal{O},\mathcal{A})$ is inconsistent if and only if $\mathcal{A} \models \varphi_M$.*

Proof. Any ABox is consistent with $\mathcal{O}_{link}$ and $\mathcal{O}_{color}$. For $\mathcal{O}_{machine}$, there is a finite number of possible violations of axioms and all of them are definable by a FO$(<)$-formula. □

Corollary. *The query Q_M is rewritable into* FO$(<)$*, linear Datalog, or Datalog, if and only if it is rewritable to the respective language over the class of ABoxes consistent with $\mathcal{O}$.*

Given that, we can now safely restrict our attention to ABoxes consistent with $\mathcal{O}$. We denote such ABoxes as $\mathcal{O}$-ABoxes.

Lemma 5. *There is an $\mathcal{O}$-ABox containing a proper computation path if and only if M halts.*

Proof. $(\Rightarrow)$ Let $\mathcal{A}$ be an $\mathcal{O}$-ABox containing a proper computation path (σ, m), $\sigma = (a_0,\dots,a_n)$. Consider a model $\mathcal{M}$ of $(\mathcal{O},\mathcal{A})$ and the corresponding path in $\mathcal{M}$. We construct a computation of M:

$$(s_0,c_0,d_0),(s_1,c_1,d_1),(s_2,c_2,d_2)\dots(s_n,c_n,d_n), \tag{36}$$

so that for each i, $0 \leqslant i \leqslant n$:

- each s_i corresponds to the unique $S_i^{\alpha\beta}$ which holds for each a_i in $\mathcal{M}$ due to the axioms (18), (23), (25);
- $c_i = m_1^i - m$ and $d_i = m_2^i - m$.

It is easy to see that (36) is a proper computation of M, and thus that M halts.

$(\Leftarrow)$ Let now M have a proper computation of the form (36). Construct an ABox $\mathcal{A}$ by allowing:

- $ind(\mathcal{A}) = \{a_i \mid 0 \leqslant i \leqslant n\}$;
- $R(a_i, a_{i+1}, 0),\ U_1(a_i, a_{i+1}, c_i),\ U_2(a_i, a_{i+1}, d_i) \in \mathcal{A},\ 0 \leqslant i < n$;

- $I(a_0, 0) \in \mathcal{A}, \ 0 \leqslant i \leqslant n$.

It is easy to see that $\mathcal{A}$ is consistent with $\mathcal{O}$, and that $((a_0, \dots, a_n), 0)$ is a proper computation path. $\square$

Let $G_{E,\mathcal{A}}(m)$ denote the set of assertions of $\mathcal{A}$, restricted to the role name E and the time point m. We view it as a (non-directed) graph, with $ind(G_{E,\mathcal{A}}(m))$ as the set of nodes and $\{(u, v) \mid E(u, v, m) \in G_{E,\mathcal{A}}(m)$ or $E(v, u, m) \in G_{E,\mathcal{A}}(m)\}$ as edges. Further, for $a \in ind(\mathcal{A})$, let $G_{E,\mathcal{A}}(a, m)$ be the connected component of $G_{E,\mathcal{A}}(m)$, containing a. We refer to the nodes of $G_{E,\mathcal{A}}(a, m)$ by $ind(G_{E,\mathcal{A}}(a, m))$.

Lemma 6. *For an arbitrary $\mathcal{O}$-ABox $\mathcal{A}$, the certain answer to Q_M over $\mathcal{A}$ is 'yes' if and only if there is a proper computation path $((a_0, \dots, a_n), m)$ in $\mathcal{A}$ such that $G_{E,\mathcal{A}}(a_n, m)$ is not 3-colorable.*

Proof. ($\Rightarrow$) Consider an $\mathcal{O}$-ABox $\mathcal{A}$ and suppose that in every model $\mathcal{M}$ of $(\mathcal{O}, \mathcal{A})$ there is an individual a, and a time point $\ell \in tem(\mathcal{A})$, such that *Yellow*$(a, \ell)$ is true in $\mathcal{M}$. Since *Yellow* $\notin \Sigma$, then $(\mathcal{O}, \mathcal{A}) \models \exists x, t.B(x, t)$. Thus, by $\mathcal{O}_{link}$, there is a proper computation path $((a_0, \dots, a_n), m)$ in $\mathcal{A}$. If $G_{E,\mathcal{A}}(a_n, m)$ is 3-colorable, then there exists a model without *Yellow*. Therefore, it is not 3-colorable.
($\Leftarrow$) Suppose that $\mathcal{A}$ contains a proper computation path $((a_0, \dots, a_n), m)$. Then, $I(a_0, m) \in \mathcal{A}$, and thus $(\mathcal{O}, \mathcal{A}) \models S_0^{00}(a_0, m)$ and $(S_f^{\alpha\beta} \sqcap A \sqcap B)(a_n, m)$, for some $\alpha, \beta \in \{0, +\}$. And therefore, $(\mathcal{O}, \mathcal{A}) \models B(b, m)$ for all $b \in ind(G_{E,\mathcal{A}}(a_n, m))$. Suppose that $G_{E,\mathcal{A}}(a_n, m)$ is not 3-colorable. Then any coloring of it with *Red, Blue, Green, Yellow* contains all four colors. Due to $\mathcal{O}_{color}$, every model $\mathcal{M}$ of the $\mathcal{O}$-ABox $\mathcal{A}$ contains an individual c, such that *Yellow*(c, m) holds in $\mathcal{M}$. I.e., the certain answer to Q_M over $\mathcal{A}$ is 'yes'. $\square$

Lemma 7. *The OMBAQ $Q_M = (\mathcal{O}, \Sigma,$ Yellow$)$ is* FO$(<)$*-rewritable if and only if M does not halt.*

Proof. ($\Rightarrow$) If M does not halt then, by Lemma 5 there is no proper computation path in any $\mathcal{O}$-ABox $\mathcal{A}$, so by Lemma 6 the certain answer to Q_M is 'no'. Then, $\varphi_M \longrightarrow \bot$ is a FO$(<)$-rewriting of Q_M.

($\Leftarrow$) Suppose now that M halts. By Lemma 5, there is an $\mathcal{O}$-ABox, $\mathcal{A}$, containing a proper computation path $(\sigma, 0)$. Let G be an arbitrary connected graph. Starting from $\mathcal{A}$, construct an $\mathcal{O}$-ABox $\mathcal{A}_G$ by adding to $\mathcal{A}$ a copy of G at 0. The edges of G are represented by the role name E. Clearly, the translation from G to $\mathcal{A}_G$ is in AC^0. Then evaluating Q_M over $\mathcal{A}_G$ answers if G is not 3-colorable. Therefore, answering Q_M is coNP-hard and thus not in AC^0. $\square$

Therefore, deciding FO$(<)$-rewritability of Q_M would solve the halting problem for M. We further observe that:

1. FO$(<)$-rewritability implies (linear) Datalog-rewritability.
2. Linear Datalog-rewritability implies answering in NLogSpace.
3. Datalog-rewritability implies answering in PTime.

Thus modulo the assumptions NLogSpace $\neq$ coNP and PTime $\neq$ coNP, respectively, deciding linear Datalog-rewritability and Datalog-rewritability of Q_M would solve the halting problem for M.

Theorem 2 for OMBAQs now follows from the above. The case for OMAQs requires a small adjustment described below.

Fix a 2-counter machine M. The construction of a corresponding OMAQ Q'_M follows that of the OMBAQ Q_M as described above. However, we can not use the a OMAQ to check for the *Yellow* label *somewhere in the ABox.*

To handle this, we, once again, make use of a fresh role name W.

Let $\mathcal{O}_{\mathit{sensor}}$ consist of axioms that make $Yellow$ spread along W in the same fashion as it was done for the concept B. Then $Q'_M(x,t) = (\mathcal{O} \cup \mathcal{O}_{\mathit{sensor}}, \Sigma \cup \{W\}, Yellow(x,t))$. If M does not halt, then Q'_M is FO($<$)-rewritable similarly to Q_M. Suppose that M halts. We adjust the reduction of (non) 3-colorability problem for the case of OMAQs. Given a graph G, first construct the ABox $\mathcal{A}_G$, described in the proof of Lemma 7. Then add a fresh individual w and use W to connect all nodes of G at time m with w. Then G is 3-colorable if and only if there is a model $\mathcal{M}$ which does not contain $Yellow(w,0)$. $\square$

Theorem 3. *Checking rewritability of DL-Lite$_{bool}$ OMAQs (OMBAQs) into* FO *is in 3NExpTime.*

Proof. The decision procedure relies on a translation of an OMAQ (OMBAQ) to an equivalent Monadic Disjunctive Datalog program and further on an existing FO-rewritability checking algorithm for MDDLog given by Feier et al. [7]. They provide a translation to MDDLog for ontologies specified in $\mathcal{ALCI}$ with simple role inclusions of type $R \sqsubseteq S$, called role subsumption axioms. Here we adapt this technique to the case of arbitrary boolean role inclusions. We start with the necessary definitions.

Fix two alphabets, of variables $Var = \{X_1, X_2, \dots\}$ and of constants $Const = \{K_1, K_2, \dots\}$. A Disjunctive Datalog (DDLog) rule is an expression of the form

$$\alpha_1(\mathbf{X_1}) \vee \cdots \vee \alpha_k(\mathbf{X_k}) \longleftarrow \beta_1(\mathbf{Y_1}) \wedge \cdots \wedge \beta_m(\mathbf{Y_m}) \tag{37}$$

Here α_i, β_j are relation symbols with associated arities and all $\mathbf{X_i}$, $\mathbf{Y_j}$ are tuples of symbols in $Var \cup Consts$ of lengths respecting the arity. We call the left hand part of the rule its *head* and the right hand part its *body.* Each variable occurring in the head must also occur in the body. A DDLog program π is a finite collection of DDLog rules with a selected relation $Goal$ that never appears in rule bodies and occurs only in "goal rules" of the form

$$Goal(\mathbf{X}) \longleftarrow \beta_1(\mathbf{Y_1}) \wedge \cdots \wedge \beta_m(\mathbf{Y_m}) \tag{38}$$

Relation symbols that occur in the head of at least one rule of π are called *IDB relations*, and all those occurring only in rule bodies are called *EDB relations.* The set of all IDB relations of π constitutes its *IDB schema*, while the set of all relation symbols of π (EDB + IDB) is its *EDB schema.*

The semantics of DDLog is borrowed from first-order logic. Denote by FO(π) a first-order theory obtained from an DDLog program π by universally quantifying its variables in all rules. Given an ABox $\mathcal{A}$ we write $\mathcal{A} \models \pi(a_1, \dots, a_n)$ if and only if FO(π) $\cup\, \mathcal{A} \models Goal(a_1, \dots, a_n)$ in the sense of the first-order logic.

Monadic DDLog (MDDLog for short) restricts IDB relations to be of arity at most 1. The size of π is the number of symbols needed to write it down, counting each relation symbol as one. A program is called *boolean* if the arity of $Goal$ is zero.

Fix a *DL-Lite$_{bool}$* OMAQ $Q(x,t) = (\mathcal{O}, \Sigma, A(x,t))$. We would like to obtain an MDDLog program π_Q that is equivalent to Q in the sense that a is a certain answer to Q over $\mathcal{A}$ if and only if $\mathcal{A} \models \pi_Q(a)$, for any Σ-ABox $\mathcal{A}$ and any individual a. If, instead, Q is an OMBAQ, then our goal is to construct a program π_Q such that the answer for Q over $\mathcal{A}$ is "Yes" if and only if $\mathcal{A} \models \pi_Q$.

The translation is similar to the one done in the proof of Th. 1. We use constants $d_{\mathbf{R}^-}$ for role types to simulate the behavior or roles in the anonymous part of the model. Constants, in fact, are just syntactic sugar: later we will show how to get rid of them.

The difference is, however, that MDDLog does not allow binary IDB relations. To capture reasoning on roles in the MDDLog encoding, we collect all possible inferences on roles as a pre-processing stage, and then project them to unary relations witnessing domain and range of roles. This step will lead to an exponential blow-up in the size of π_Q.

We now present the MDDLog encoding, π_Q, obtained by encoding the TBox in the rule set $\pi_{\mathcal{T}}$, the RBox in the rule set $\pi_{\mathcal{R}}$, and the query in the goal rule π_G. Fix an OMAQ $Q(x,t) = (\mathcal{T} \cup \mathcal{R}, \Sigma, A(x,t))$ (or an OMBAQ $Q = (\mathcal{T} \cup \mathcal{R}, \Sigma, A)$). In the following, for every concept C, by C in Q we mean that C is used in Q, and for every role R, by R in Q we mean that either R or R^- is used in Q. Recall that a role R can be both a role name P or its inverse P^-. Recall also the definition of a role type from the proof of the Theorem 1.

The EDB schema of an MDDLog program π_Q includes the following relation symbols:

- For each concept name A in Q, a unary relation $A(X)$;
- For each concept $\exists R$ in Q, a unary relation $ER(X)$;
- For each role R in Q, a binary relation $R(X,Y)$, two unary relations $ER(X), ER^-(X)$. If $\bot$ is used in role inclusions of Q, we treat $E\bot, E\bot^-$ simply as $\bot$.
- For each role type $\mathbf{R}$ of roles in Q, a constant $d_{\mathbf{R}^-}$.

Let $\pi_{\mathcal{T}}$ be the MDDLog rule set encoding the TBox, $\mathcal{T}$, such that, for each concept inclusion, $C_1 \sqcap \dots \sqcap C_k \sqsubseteq C_{k+1} \sqcup \dots \sqcup C_{k+m}$, it contains a rule:

$$D_{k+1}(X) \vee \dots \vee D_{k+m}(X) \longleftarrow D_1(X) \wedge \dots \wedge D_k(X), \tag{39}$$

where $D_i = A_i$, if $C_i = A_i$, and $D_i = ER$, if $C_i = \exists R$.
To encode the RBox, we use a projection of roles, which are binary, to pairs of monadic concepts. We first introduce some definitions.

Let $\mathrm{Sat}(\mathbf{W})$ denote the collection of all role types $\mathbf{R}$ such that $\mathbf{W} \subseteq \mathbf{R}$. For each set of roles $\mathbf{W} = \{W_1, \dots, W_t\}$ which is a role type consider the following rule (cf. (13)):

$$\left(\bigvee_{\mathbf{R} \in \mathrm{Sat}(\mathbf{W})} \mathcal{E}_{\mathbf{R}}(X) \wedge \mathcal{E}^-_{\mathbf{R}}(Y) \right) \longleftarrow W_1(X,Y) \wedge \dots \wedge W_t(X,Y) \tag{40}$$

While the head of this "rule" is in disjunctive normal form, we obtain an MDDLog rule set, denoted as $\pi^{\mathbf{S}}$, by converting (40) to a conjunctive normal and then adding a separate rule for each disjunctive clause. Furthermore, for each role W in Q consider the following rules:

$$\left(\bigvee_{\mathbf{R}\in\mathrm{Sat}(W)} \mathcal{E}_{\mathbf{R}}(X) \wedge \mathcal{E}^{-}_{\mathbf{R}}(d_{\mathbf{R}^-})\right) \longleftarrow EW(X) \tag{41}$$

The MDDLog rule set, denoted as $\pi^{\exists S}$, is obtained from the above formulas. Intuitively, this program is needed to capture properties of the anonymous part encoded via the $d_{\mathbf{R}^-}$ constants.

We can now define the MDDLog rule set, $\pi_{\mathcal{R}}$, to be the union of the following rule sets:

- $\pi^{\mathbf{R}}$, for each set of roles $\mathbf{R}$ as defined in (40);
- programs $\pi^{\exists R}$, for each role R in Q as defined in (41);
- π^{R}, which, for each role R in Q, is the following rule:

$$ER(X) \longleftarrow R(X,Y); \tag{42}$$

Finally, the goal rule, π_G, contains the rule:

$$Goal(X) \longleftarrow A(X) \qquad \text{or} \qquad Goal \longleftarrow A(X), \tag{43}$$

for Q being an OMAQ or an OMBAQ, respectively. Recall that we defined $\pi_Q = \pi_{\mathcal{T}} \cup \pi_{\mathcal{R}} \cup \pi_G$. We can now prove the following main result.

Lemma 8. *Let Q be a DL-Lite$_{bool}$ OMAQ or OMBAQ. There exists an equivalent MDDLog program π_Q of size $O\left(2^{|Q|}\right)$.*

Proof. We consider the more difficult case of OMBAQs. Let $Q = (\mathcal{O}, \Sigma, A)$ be an OMBAQ with $\mathcal{O} = \mathcal{T} \cup \mathcal{R}$, and π_Q an MDDLog program obtained from Q as described above. It is easy to check that $|\pi_Q| = O\left(2^{|Q|}\right)$.

Fix an arbitrary Σ-ABox $\mathcal{A}$. Recall that FO(π_Q) is the first-order theory obtained by universally quantifying rules of π_Q. It suffices to prove that there exists a model of $(\mathcal{O} \cup \{A \sqsubseteq \bot\}, \mathcal{A})$ if and only if there exists a model of FO(π_Q) $\cup\, \mathcal{A} \cup \{\neg\exists x.A(x)\}$.

($\Rightarrow$) Let $\mathcal{M} = \left(\Delta, \cdot^{\mathcal{M}}\right)$ be a model of $(\mathcal{O} \cup \{A \sqsubseteq \bot\}, \mathcal{A})$. We construct a model $\mathcal{N} = \left(\Delta, \cdot^{\mathcal{N}}\right)$ for FO(π_Q) $\cup\, \mathcal{A} \cup \{\neg\exists x.A(x)\}$ using the same domain Δ. For every $a \in ind(\mathcal{A})$ we set $a^{\mathcal{N}} = a^{\mathcal{M}}$. For every role type $\mathbf{R}$ choose any $d \in \left\{u \in \Delta \mid u \in (\exists R)^{\mathcal{M}} \text{ for all } R \in \mathbf{R}\right\}$. If this set is empty, take any $d \in \Delta$. We set $d^{\mathcal{N}}_{\mathbf{R}^-} = d$. Finally, for each atomic concept A set $A^{\mathcal{N}} = A^{\mathcal{M}}$ and for each role R set $R^{\mathcal{N}} = R^{\mathcal{M}}$ and $ER^{\mathcal{N}} = (\exists R)^{\mathcal{M}}$. It can be easily verified that $\mathcal{N}$ is an A-countermodel of FO(π_Q) $\cup\, \mathcal{A}$.

($\Leftarrow$) Conversely, suppose $\mathcal{N} = (\Delta, \cdot^{\mathcal{N}})$ is a model of FO(π_Q) $\cup\, \mathcal{A} \cup \{\neg\exists x.A(x)\}$. We can choose $\mathcal{N}$ to be a Herbrand model, so that $\Delta = ind(\mathcal{A}) \cup \{d_{\mathbf{R}^-} \mid \mathbf{R} \text{ is a role type } Q\}$, with an obvious interpretation $a^{\mathcal{N}} = a$ for $a \in ind(\mathcal{A})$ and $d^{\mathcal{N}}_{R} = d_{\mathbf{R}^-}$ for any $\mathbf{R}$.

We construct a model $\mathcal{M} = (\Delta, \cdot^{\mathcal{M}})$ for $(\mathcal{O} \cup \{A \sqsubseteq \bot\}, \mathcal{A})$ using the same domain Δ. We also borrow the interpretation of ABox individual names. Further, for each atomic concept A set $A^{\mathcal{M}} = A^{\mathcal{N}}$.

The interpretation of roles is defined in three steps. First, for each role R and any $u, v \in \Delta$ we let $(u, v) \in R^{\mathcal{M}}$ if $\mathcal{N} \models R(u, v)$. This way we satisfy the role assertions of the ABox.

Second, we ensure that ER predicates in $\mathcal{N}$ correspond to $\exists R$ in $\mathcal{M}$. Consider any u such that $u \in EW^{\{}\mathcal{N}\}$ for some role W. By (41), there exists a role type $\mathbf{R}$ such that $\mathcal{E}_{\mathbf{R}}(u) \wedge \mathcal{E}^{-}_{\mathbf{R}}(d_{\mathbf{R}^-})$ is true in $\mathcal{N}$. We set $(u, d_{\mathbf{R}^-}) \in R^{\mathcal{M}}$ for each $R \in \mathbf{R}$.

Third, we ensure that roles of $\mathcal{M}$ satisfy the RBox $\mathcal{R}$. For this we "recover" the inferred roles from their encoding in $\mathcal{N}$ with pairs of ER, ER^- predicates. For a pair $u, v \in \Delta$ define $\mathrm{Rol}(u, v)$ to be the set of those roles R for which we have already established $(u, v) \in R^{\mathcal{M}}$ on the previous steps. Our goal is to enrich this set so that it forms a role type.

Note, that if at least one of u and v is anonymous, $\mathrm{Rol}(u, v)$ is a role type by construction. So suppose $u, v \in ind(\mathcal{A})$. Then by (40) there exists a role type $\mathbf{R}$ such that $\mathcal{E}_{\mathbf{R}}(u) \wedge \mathcal{E}^{-}_{\mathbf{R}}(v)$ is true in $\mathcal{N}$. Finally, we set $(u, v) \in R^{\mathcal{M}}$ for all $R \in \mathbf{R}$.

Now, for every two objects $u, v \in \Delta$ the set $\{R \mid (u, v) \in R^{\mathcal{M}}\}$ is a role type. Therefore, $\mathcal{M} \models \mathcal{R}$.

It remains to verify that for each role R we have $\exists R^{\mathcal{M}} = ER^{\mathcal{N}}$. Indeed, for each $u \in \Delta$, if $ER(u)$ holds in $\mathcal{N}$ then we have $R(u, d_{\mathbf{R}^-})$ in $\mathcal{M}$ for some type $\mathbf{R} \ni R$. Therefore $u \in \exists R^{\mathcal{M}}$ by the semantics of description logic. Conversely, suppose that $R(u, v)$ holds in $\mathcal{M}$. If $R(u, v) \in \mathcal{A}$, then we have $ER(u), ER^-(v)$ in $\mathcal{N}$ by the rule (42). If, on the other hand, $R(u, v)$ was added to $\mathcal{M}$ on the second or the third step of role interpretation, then by construction $ER(u)$ and $ER^-(v)$ must be true in $\mathcal{N}$.

Therefore, the interpretation of concepts in $\mathcal{M}$ coincides with the corresponding unary predicates of $\mathcal{N}$. Since $\mathcal{N}$ satisfies rules of type (39), $\mathcal{M} \models \mathcal{T}$. Therefore, $\mathcal{M}$ is a model of $(\mathcal{O} \cup \{A \sqsubseteq \bot\}, \mathcal{A})$. □

We note that $d_{\mathbf{R}^-}$ constants can be eliminated by a cost of a polynomial growth of the program. Indeed, let π^c_Q comprise those rules of π_Q that use constants, while π^{cf}_Q consist of all constant-free rules. Let also π^g_Q be the result of grounding *all* rules in π_Q with constants $\{d_{\mathbf{R}} \mid R \text{ is a role type}\}$. Since each rule contains at most two variables, the size of π^g_Q is polynomial in the size of π_Q. Finally, it is not hard to see that $\pi^g_Q \cup \pi^{cf}_Q$ is a constant-free program equivalent to π_Q.

The program $\pi^g_Q \cup \pi^{cf}_Q$ can be further converted into an instance of generalized coCSP of size triple exponential in the size of the original query Q (cf. Feier et al. [7], Theorem 3.3, which is restated from Feder and Vardi [25]). In turn, FO-rewritability for coCSP is NP-complete (see Larose et al. [26], Barto [27], Chen and Larose [28]). □