Evaluating Conjunctive and Graph Queries over the EL Profile of OWL 2

Giorgio Stefanoni
St Catherine's College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2015

A Frieda ed Enrico, i miei genitori.

Abstract

OWL 2 EL is a popular ontology language that is based on the $\mathcal{EL}$ family of description logics and supports regular role inclusions—axioms that can capture compositional properties of roles such as role transitivity and reflexivity. In this thesis, we present several novel complexity results and algorithms for answering expressive queries over OWL 2 EL knowledge bases (KBs) with regular role inclusions.

We first focus on the complexity of conjunctive query (CQ) answering in OWL 2 EL and show that the problem is PSpace-complete in combined complexity, the complexity measured in the total size of the input. All the previously known approaches encode the regular role inclusions using finite automata that can be worst-case exponential in size, and thus are not optimal. In our PSpace procedure, we address this problem by using a novel, succinct encoding of regular role inclusions based on pushdown automata with a bounded stack. Moreover, we strengthen the known PSpace lower complexity bound and show that the problem is PSpace-hard even if we consider only the regular role inclusions as part of the input and the query is acyclic; thus, our algorithm is optimal in knowledge base complexity, the complexity measured in the size of the KB, as well as for acyclic queries.

We then study graph queries for OWL 2 EL and show that answering positive, converse-free conjunctive graph queries is PSpace-complete. Thus, from a theoretical perspective, we can add navigational features to CQs over OWL 2 EL without an increase in complexity.

Finally, we present a practicable algorithm for answering CQs over OWL 2 EL KBs with only transitive and reflexive composite roles. None of the previously known approaches target transitive and reflexive roles specifically, and so they all run in PSpace and do not provide a tight upper complexity bound. In contrast, our algorithm is optimal: it runs in NP in combined complexity and in PTime in KB complexity. We also show that answering CQs is NP-hard in combined complexity if the query is acyclic and the KB contains one transitive role, one reflexive role, or nominals—concepts containing precisely one individual.

Acknowledgements

This thesis would not have been possible without the help of my supervisors Prof. Boris Motik and Prof. Ian Horrocks. I am particularly grateful to Boris for always being patient and never loosing faith in me, in spite of my many oddities. I thank Ian for making it possible for me to come to Oxford in the first place, and for always being there for me when I needed him. Furthermore, I wish to express my sincere gratitude to two exceptional examiners, Prof. Frank Wolter and Dr Egor V. Kostylev.

I would also like to thank all the members of the KRR group. I can't name you all, but because of you each one of my working days at the department has been remarkable.

I am also very grateful to St. Catherine's College for providing a warm environment where I met so many wonderful people that I now can proudly call my friends.

I would like to extend my gratitude to the people in the KRDB group at the Free University of Bolzano, and the people in the KBS group at the Technische Universität Wien, who have inspired my interest in the field of knowledge representation.

I wish to thank all my friends and relatives in Bolzano, who have always supported me and cared for me. I cannot put into words how much I am grateful to my parents: all that I have achieved would not have been possible without your support.

I have had the fortune to share the journey that has led to this thesis with Marta. *Amore*, thank you for caring so much about me and for always making me smile. I feel extremely lucky to have you in my life.

My doctorate was supported by a joint studentship by the Engineering and Physical Sciences Research Council (EPSRC) and Alcatel-Lucent Labs. I am particularly grateful to Peter, Chris, and Aidan from Alcatel-Lucent who have taken good care of me during my visits in Dublin.

Contents

Chapter 1

Introduction

The *Web Ontology Language* OWL 2 is the W3C standard for authoring ontologies on the Web [23]. OWL 2 ontologies represent a domain of discourse by describing the relevant entities and the relationships that exist among them, and they have been successfully applied in modelling domains in a variety of disciplines such as biomedicine [91], geography [42], astronomy [27], agriculture [69], computer security [104], web services [57], and e-science [49]. One of the defining characteristics of OWL 2 is that it is formally underpinned by description logics (DLs) [6], a family of logic-based knowledge representation languages.

1.1 Representing Knowledge in Description Logics

Description logics describe a domain of interest using three types of entities—*individuals*, *atomic concepts*, and *roles*—that correspond to constants, unary predicates, and binary predicates of first-order logic, respectively. Individuals provide names for objects in the domain, atomic concepts represent sets of objects, and roles represent binary relationships among objects; for example, the individual frieda might be used to represent a particular person whose name is Frieda; the concept Woman might be used to represent the set of all women, and the role isMotherOf might be used to represent the binary relationship between mothers and their children.

Description logic *knowledge bases* (KBs) correspond to OWL 2 ontologies, and model a domain of interest by relating individuals, concepts, and roles using *logical axioms*. In

a knowledge base, one usually distinguishes between three types of axioms: assertional (ABox) axioms, terminological (TBox) axioms, and relational (RBox) axioms. Each type of axiom describes a different aspect of the domain of interest.

Assertional axioms describe the known properties of individuals in the domain using *assertions*. For example, we can state that the individual frieda is the mother of the individual giorgio, and that frieda is a woman using the following role and concept assertions.

$$\mathsf{isMotherOf}(\mathsf{frieda}, \mathsf{giorgio}) \tag{1.1}$$

$$\mathsf{Woman}(\mathsf{frieda}) \tag{1.2}$$

Terminological axioms relate different concepts using *concept inclusions*. For example, we can state that every woman is a human using the following concept inclusion.

$$\mathsf{Woman} \sqsubseteq \mathsf{Human} \tag{1.3}$$

Relational axioms complement TBox axioms by relating different roles using *role inclusions*. For example, we can state that, if x is the mother of y, then x is a parent of y using the following role inclusion.

$$\mathsf{isMotherOf} \sqsubseteq \mathsf{isParentOf} \tag{1.4}$$

In contrast to other knowledge representation formalisms such as semantic networks [89] and frame-based systems [76], each DL/OWL knowledge base has a well-defined semantics. Hence, one can infer implicit information from the explicit axioms in the knowledge base using principled logical reasoning. For example, we can combine assertion (1.2) and axiom (1.3) to infer that frieda is a human.

Reasoning is the act of computing information that logically follows from the axioms in a knowledge base. Three standard reasoning tasks have historically received particular attention from the DL community: *consistency checking* amounts to checking whether the axioms in the knowledge base are not contradictory, *subsumption checking* amounts to determining whether one concept is more general than another concept, and *instance*

checking amounts to checking whether an individual is an instance of a concept.

In their seminal work, Levesque and Brachman [68] identified a trade-off between the expressivity of a description logic and the complexity of reasoning with it. Since then, a substantial body of research was dedicated to classifying description logics based on the complexity of their reasoning tasks as well as on developing automatic tools that can efficiently solve these problems.

1.2 The Profiles of OWL 2

OWL 2 is a family of ontology languages that originates from the large body of research in description logics. The most prominent member of the family, OWL 2 DL, is logically underpinned by the expressive description logic $\mathcal{SROIQ}$ [53]. The expressivity of OWL 2 DL is not only reflected by the number of complex domains this ontology language can capture, but also by the complexity of the associated reasoning tasks, which are doubly exponential in knowledge base size.

Since many application domains can be represented using only a small fragment of the features offered by OWL 2 DL, the W3C standard complements OWL 2 DL with three sublanguages: the RL, the QL, and the EL profiles of OWL 2. All these profiles share a common design rationale: they trade expressive power in favour of tractability of reasoning.

The RL profile of OWL 2 is inspired by research in combining description logics with rule-based knowledge representation languages [43], and its defining characteristic is that reasoning can be implemented using optimised techniques from deductive databases [1]. The QL profile of OWL 2 is based on the DL-Lite family of DLs [14], and it can capture many of the features available in popular conceptual models such as UML class diagrams and ER diagrams. Although their expressivity is sufficient in many practical applications, the RL and QL profiles cannot capture large and widely used biomedical knowledge bases, such as the Gene Ontology[1], the National Cancer Institute thesaurus[2], and SNOMED CT [91]. These prominent knowledge bases, however, can be captured by the EL profile of OWL 2, which is based on the $\mathcal{EL}$ family of DLs [4] and was designed to allow for representing

[1] http://geneontology.org
[2] http://ncit.nci.nih.gov

complex domains while ensuring tractability of the standard reasoning tasks.

1.3 Querying Description Logic Knowledge Bases

In recent years, DLs and OWL 2 have been steadily gaining in popularity because they provide the developers of modern information systems with a flexible graph-like data model that is convenient in application areas such as the Semantic Web [45], social network analysis [32], and network traffic analysis [8], to name just a few. In addition to the standard reasoning tasks, in these data-intensive applications it is natural to consider more expressive languages for querying DL knowledge bases such as the languages commonly used to query relational and graph databases. In particular, answering queries over DL knowledge bases has been the core reasoning task in applications as diverse as monitoring financial products within the Italian Ministry of Economy and Finance [25], accessing real-time diagnostic data of turbines [35], and integrating configuration data of air traffic control systems [16].

While queries over relational and graph databases are answered only against the explicitly stated assertions, queries over DL knowledge bases are answered by taking into account the explicit assertions plus the assertions that can be inferred using the terminological and relational axioms in the knowledge base. Hence, query answers that implicitly follow from the knowledge base can be made explicit using suitably tailored query answering algorithms, thus potentially addressing incompleteness in the data. For example, the query asking for each object that is human and is the parent of someone will return an empty result over the database containing assertions (1.1) and (1.2), but will return frieda over the knowledge base that complements these two assertions with axioms (1.3) and (1.4).

Motivated by the practical interest in developing efficient query answering systems for DL knowledge bases, in recent years a lot of work has been focused on determining the computational properties of query answering over DL knowledge bases. In the literature, instead of studying the complexity of computing all the query answers that are entailed by a knowledge base, it has become common to study the associated decision problem called the *recognition problem*, which decides whether a given query answer can be inferred from the knowledge base; in other words, whether the given answer is a certain answer to the

query over the knowledge base. Moreover, the size of the query is typically several order of magnitude smaller than the size of the knowledge base, in particular of the size of the ABox; thus, to analyse the influence that various parts of the input have on the complexity of the problem, one customarily studies three complexity measures: *combined complexity* measures the complexity in terms of the combined size of the query and the KB, *knowledge base complexity* measures the complexity in terms of the size of the KB (i.e., the query is considered to be fixed), and *data complexity* measures the complexity in terms of the size of the assertions (i.e., the query, the TBox and RBox axioms are considered to be fixed).

In contrast to relational and graph databases where the complexity of the query answering problem is determined uniquely by the expressivity of the query language at hand, the computational properties of query answering over DL knowledge bases depend on the expressivity of both the constructs used in the knowledge base and the query language used. A variety of query languages have been considered for querying description logics knowledge bases and they can be classified in two main groups depending on whether they originate from relational or graph databases.

1.3.1 Relational Query Languages

Languages for querying relational databases are centred around the class of *first-order queries* [1]. While first-order queries provide the foundations for SQL and can be efficiently answered over relational databases, they permit asking for negative information. For example, a first-order query can ask for all the objects in a knowledge base that are human but are not women. Unfortunately Rosati [87] and Gutiérrez-Basulto et al. [46] have shown that negation in the query language easily causes the undecidability of the query answering problem, even for inexpressive description logics.

To regain decidability, one usually considers positive fragments of first-order queries, in particular *conjunctive queries* (CQs) have received a lot of attention as they correspond to the well-known select-project-join fragment of SQL. For expressive knowledge bases formulated in OWL 2 DL, CQ answering is at least exponential in combined and knowledge base complexities [36, 71] and intractable in data complexity [13, 82], thus suggesting that efficiently answering CQs can be challenging over large OWL 2 DL knowledge bases.

In contrast, the data complexity of CQ answering becomes tractable for all the profiles of OWL 2, which suggests that conjunctive queries can be efficiently answered even over large knowledge bases. In addition, for the RL and the QL profiles of OWL 2, many worst-case optimal, highly optimised CQ answering algorithms have been proposed and have been shown to work well in practice [14, 85, 99, 60, 83, 37, 101, 79].

1.3.2 Navigational Query Languages

Relational query languages cannot express recursive properties such as reachability [1], and so their expressivity is often insufficient in applications that exploit the graph-like data model underlying OWL 2 knowledge bases and thus require exploring connections in graphs. As the popularity of graph databases is on the rise, a number of navigational languages for querying graph-like data have been proposed.

Regular path queries (RPQs) [22, 7] are capable of describing connections between graph vertices using regular expressions [56], allowing users to 'navigate' inside a graph. For example, the RPQ

$$\mathsf{isPartOf}^* \cdot \mathsf{hasLocation} \tag{1.5}$$

retrieves all pairs of vertices connected via zero or more isPartOf edges followed by one hasLocation edge.

Inspired by the XPath query language for XML [26], *graph XPath queries* (GXQs) have been recently proposed for querying graph databases [70] and DL knowledge bases [61, 11]. They extend RPQs by allowing for backward navigation using the *converse operator*, negation on regular expressions, and checking properties of vertices using Boolean combinations of *node tests*—that is, concepts or existential quantifications over paths. For example, the graph XPath query

$$\mathsf{isPartOf}^* \cdot \mathsf{test}(\mathsf{Cell} \wedge \neg\langle \mathsf{hasSpeciality} \rangle) \cdot \mathsf{hasLocation} \tag{1.6}$$

refines (1.5) by requiring that the node between the isPartOf edges and the hasLocation edge satisfies the concept Cell and does *not* have an outgoing hasSpecialty edge.

While the computational properties of RPQs have been extensively studied for the members of the OWL 2 family [15, 83, 9, 17], there are comparatively fewer works on studying the complexity of graph XPath queries over DL knowledge bases. Kostylev et al. [61] observed that GXQs are closely related to propositional dynamic logic with full negation [48], which shows that answering GXQs over DL knowledge bases is undecidable even with respect to the empty knowledge base. Several GXQ fragments were proposed as a possible solution to this problem, among them *positive GXQs* prohibit negation over path expressions and concepts. Answering positive GXQs is tractable in data complexity for the RL, QL, and EL profiles of OWL 2 [70, 11, 61], which makes positive GXQs an appealing language for querying the profiles of OWL 2.

1.4 Querying the EL Profile of OWL 2

Motivated by the popularity of the EL profile of OWL 2 in important disciplines such as biology and medicine, and by the increasing interest in answering expressive queries over OWL 2 knowledge bases, in this thesis we study the problem of answering conjunctive and graph queries over OWL 2 EL knowledge bases.

One of the reasons for the popularity of the EL profile of OWL 2 is that it supports role inclusions of the form $R_1 \cdots R_n \sqsubseteq R$, sometimes called *complex role inclusions*, that can express both the hierarchical and the compositional properties of roles. These axioms are useful in many applications because they allow for axiomatising part-whole relations—for example, the complex role inclusions in (1.7)–(1.9) state that $\mathsf{isPartOf}$ is a reflexive and transitive role, and that, if x is located in y and y is strictly part of z, then x is part of z.

$$\epsilon \sqsubseteq \mathsf{isPartOf} \tag{1.7}$$

$$\mathsf{isPartOf} \cdot \mathsf{isPartOf} \sqsubseteq \mathsf{isPartOf} \tag{1.8}$$

$$\mathsf{hasLocation} \cdot \mathsf{isStrictPartOf} \sqsubseteq \mathsf{isPartOf} \tag{1.9}$$

While they are an important distinguishing feature of OWL 2 EL, complex role inclusions have already been singled out as a source of undecidability in expressive description

logics prior to the introduction of the $\mathcal{EL}$ family of DLs. Complex role inclusions loosely correspond to context-free grammars: if each role inclusion $R_1 \cdots R_n \sqsubseteq R$ in a knowledge base is seen as a production rule $R \to R_1 \cdots R_n$, then the knowledge base induces a context-free language $\mathcal{L}(R)$ for each role R. Using this correspondence, Wessel [105] showed that checking consistency in knowledge bases formulated in a fragment of OWL 2 DL with complex role inclusions is undecidable. To regain decidability, Horrocks and Sattler [51] proposed a syntactic *regularity restriction* on role inclusions ensuring that each language $\mathcal{L}(R)$ is regular and can thus be recognised using a finite automaton (FA). The OWL 2 DL profile of OWL 2 supports complex role inclusions and thus incorporates an extended version of the regularity restriction by Horrocks and Sattler [51] into its definition.

Even with unrestricted role inclusions, all standard reasoning problems for members of the $\mathcal{EL}$ family of DLs can be solved in polynomial time [4]. Using the correspondence between role inclusions and context-free grammars, however, Krisnadhi and Lutz [63], Krötzsch et al. [66], and Rosati [86] independently proved that answering CQs over $\mathcal{EL}$ knowledge bases with unrestricted role inclusions is undecidable; furthermore, Krötzsch et al. [66] also showed that checking concept subsumptions over $\mathcal{EL}$ knowledge bases with inverse roles and unrestricted role inclusions is undecidable.

OWL 2 EL inherits the regularity restriction from OWL 2 DL, and so the undecidability proofs of CQ answering do not apply to OWL 2 EL. In fact, Krötzsch et al. [66] showed that answering CQs over $\mathcal{EL}$ knowledge bases extended with regular role inclusions is PSPACE-hard in combined complexity, and they proposed a CQ answering algorithm for a fragment of OWL 2 EL with regular role inclusions, but without reflexive roles. This algorithm, however, runs in PSPACE in combined and knowledge base complexities only if, for each role R, language $\mathcal{L}(R)$ can be represented using an automaton of polynomial size; however, Kazakov [54] showed that, in some cases, the size of this automaton is necessarily exponential in the size of the knowledge base. Hence, the algorithm by Krötzsch et al. [66] does not provide us with a matching PSPACE upper bound for the problem. Ortiz et al. [83] proposed a different algorithm for answering CQs over OWL 2 EL knowledge bases (with regular role inclusions and without any restriction on the usage of other features). Similarly to the approach by Krötzsch et al. [66], the algorithm by Ortiz et al. [83] also encodes regular role

inclusions using finite automata; moreover, both procedures use automata techniques that are not practicable due to extensive don't-know nondeterminism. Hence, while both of these algorithms run in time polynomial in the size of the data and thus settle the question of data complexity, they do not settle the question of knowledge base and combined complexities and do not provide a practicable approach to answering conjunctive queries over OWL 2 EL knowledge bases.

When the complex role inclusions in an OWL 2 EL knowledge bases express only transitive and reflexive roles, each language $\mathcal{L}(R)$ induced by the knowledge base can be recognised using finite automaton of polynomial size. Thus, the algorithm by Krötzsch et al. [66] provides a PSPACE upper bound for CQ answering over $\mathcal{EL}$ knowledge bases with transitive roles. This algorithm, however, does not handle this restricted form of complex role inclusions specifically, so it is not clear whether the upper bound is optimal for combined and knowledge base complexities. This is interesting because transitive roles are a known source of complexity of CQ answering over knowledge bases [28, 36, 39] and, together with reflexive roles, suffice to express simple graph properties such as reachability, thus bridging the gap between conjunctive and navigational queries. For example, in the knowledge base that contains role inclusions (1.7) and (1.8), the RPQ in (1.5) can be equivalently expressed as the following conjunctive query which asks for all pairs of objects x and z for which an object y exists such that x is part of y, and y is located in z.

$$q = \exists y.\, \mathsf{partOf}(x, y) \wedge \mathsf{hasLocation}(y, z) \tag{1.10}$$

1.5 Contributions

In this thesis, we present several novel complexity results and algorithms for answering conjunctive and graph queries over knowledge bases formulated in (fragments of) OWL 2 EL. All our algorithms are worst-case optimal in combined, knowledge base, and data complexities and are based on a number of original results, which we summarise next.

In Chapter 5, we present a translation of OWL 2 EL knowledge bases into datalog programs. Our translation can be applied to any fragment of OWL 2 EL, and the translation

preserves both knowledge base consistency and answers to instance queries—conjunctive queries that consist of a single unary atom. Moreover, the models of the datalog program are models of the knowledge base, and so each answer to a CQ over the knowledge base is also an answer to the CQ over the datalog program. In contrast, the converse does not necessarily hold, so evaluating conjunctive queries over the datalog program may produce *unsound answers*.

Based on our datalog translation, in Chapter 6 we present a procedure for answering CQs over $\mathcal{ELHO}_\bot^+$ knowledge bases—the fragment of OWL 2 EL in which complex role inclusions express only transitive and reflexive roles. Our algorithm runs in nondeterministic polynomial time in combined complexity, but is tractable in data and knowledge base complexities; thus, we close the long-standing open questions about the combined and knowledge base complexities of CQ answering in $\mathcal{EL}$ variants with transitive roles. Our procedure generalises the combined approach for the $\mathcal{ELH}_\bot$ fragment of OWL 2 EL by Lutz et al. [72]. To answer a CQ, we evaluate the query over the datalog translation of the knowledge base to obtain *candidate answers*, and then we filter out unsound candidate answers using an external filtering procedure. Our filtering procedure runs in nondeterministic polynomial time, and we prove that this is worst-case optimal—that is, checking whether a candidate answer is sound is an NP-hard problem. To obtain a goal-directed filtering procedure, we developed optimisations that reduce the number of nondeterministic choices. Finally, our filtering procedure runs in NP only for candidate answers that depend on both existential information in the knowledge base, and transitive or reflexive roles. In other words, our filtering procedure has pay-as-you-go behaviour—in particular, filtering can be done in polynomial time for knowledge bases formulated in the fragment of $\mathcal{ELHO}_\bot^+$ obtained by disallowing transitive and reflexive roles.

To experimentally validate the feasibility of our method for answering CQs over $\mathcal{ELHO}_\bot^+$ knowledge bases, we have implemented a new query answering system called EOLO. In Chapter 7, we present the results of our preliminary evaluation which suggests that, while some queries can be challenging, our approach provides a practical basis for answering conjunctive queries over a large fragment of OWL 2 EL.

Towards obtaining a worst-case optimal CQ answering algorithm for OWL 2 EL, in

Chapter 8 we present a novel succinct encoding of the languages induced by regular role inclusions using pushdown automata (PDAs)—that is, FAs extended with a stack. We show that, for each role R, we can construct in polynomial time a PDA that accepts language $\mathcal{L}(R)$ and whose computations use a stack of size linear in the number of role inclusions in the knowledge base. Bounded-stack PDAs [3] recognise precisely the class of regular languages and can be exponentially more succinct than finite automata [34]. Apart from allowing us to obtain a worst-case optimal CQ answering algorithm for OWL 2 EL, the tableau algorithm by Horrocks et al. [53] used in popular reasoners for OWL 2 DL such as Pellet [93] and FaCT++ [98] can be straightforwardly modified to use bounded-stack PDAs instead of FAs, which could eliminate a potential source of inefficiency in practice.

Building upon the succinct encoding of regular role inclusions based on PDAs, in Chapter 9 we present a procedure for answering conjunctive queries over $\mathcal{ELRO}_\bot^+$ knowledge bases—the DL logically underpinning OWL 2 EL. Our algorithm runs in PSPACE in both combined and knowledge base complexities, but is tractable in data complexity; thus, we also close the open question about the complexity of answering CQs over OWL 2 EL knowledge bases. The procedure extends and refines the CQ answering algorithm for $\mathcal{ELHO}_\bot^+$ from Chapter 6. Regular role inclusions, however, increase the complexity of the filtering step: unlike the filtering procedure for $\mathcal{ELHO}_\bot^+$, the filtering procedure for $\mathcal{ELRO}_\bot^+$ runs in PSPACE. This is worst-case optimal as Krötzsch et al. [66] showed that CQ answering over $\mathcal{ELRO}_\bot^+$ knowledge bases is PSPACE-hard. Furthermore, we show that our algorithm runs in nondeterministic polynomial time if the role inclusions are considered fixed, and in polynomial time if both the query and the role inclusions are fixed, which suggests that role inclusions are the main source of complexity in answering CQs over OWL 2 EL.

Motivated by the high combined complexity of answering CQs over OWL 2 EL knowledge bases, in Chapter 10 we investigate the computational properties of *acyclic queries*—an important class of Boolean CQs for which query answering is tractable in relational databases [106]. We start by introducing a new class of *arborescent queries*—tree-shaped acyclic CQs in which all roles point towards the parent; then, we improve the known PSPACE lower bound by Krötzsch et al. [66] and show that CQ answering over $\mathcal{ELRO}_\bot^+$ knowledge

bases is PSPACE-hard even when the query is arborescent and just the role inclusions are considered as part of the input (i.e., the query and all other parts of the knowledge base are fixed). This novel result thus shows that our algorithm from Chapter 9 is optimal also in knowledge base complexity. Moreover, we show that answering arborescent queries is NP-hard even for very simple knowledge bases that contain either a single transitive role or a single reflexive role, which shows that complex role inclusions immediately cause the intractability of acyclic query answering over OWL 2 EL knowledge bases. This is interesting because Bienvenu et al. [10] showed that answering acyclic queries is tractable over the $\mathcal{ELH}_\bot$ fragment of OWL 2 EL obtained by disallowing complex role inclusions and *nominals*, concepts that contain precisely one individual. Finally, we show that, although answering arborescent queries is tractable over the fragment of OWL 2 EL with nominals but without complex role inclusions, the problem becomes intractable for arbitrary acyclic queries. Hence, our results suggest that $\mathcal{ELH}_\bot$ is the maximal member of the $\mathcal{EL}$ family of DLs for which acyclic CQ answering is tractable.

In Chapter 11, we study the complexity of graph XPath queries and their extensions to conjunctive graph XPath queries (CGXQ) over $\mathcal{ELRO}^+_\bot$ knowledge bases, and we show that positive, converse-free CGXQs can be answered over $\mathcal{ELRO}^+_\bot$ knowledge bases in PSPACE using the CQ answering algorithm from Chapter 9. In particular, $\mathcal{ELRO}^+_\bot$ (and thus also OWL 2 EL) supports role inclusions, self restrictions, and reflexive roles, and we use these expressive features to polynomially reduce answering a CGXQ to answering a CQ over an extended knowledge base. We also show that answering positive, converse-free GXQs can be done in time polynomial in the input size. These results are interesting because Bienvenu et al. [11] proved that answering positive GXQs over $\mathcal{ELH}_\bot$ knowledge bases is EXPTIME-complete; hence, adding the converse operator increases the complexity of graph XPath queries. Our results thus show that answering GXQs and CGXQs is as difficult as instance checking and answering conjunctive queries, respectively which, at least from a theoretical perspective, makes positive, converse-free GXQs and CGXQs appealing as navigational query languages for OWL 2 EL knowledge bases.

Finally, in Chapter 12 we analyse the different syntactic conditions that have been

proposed to ensure the regularity of the languages induced by OWL 2 knowledge bases with complex role inclusions. Even though the original regularity condition by Horrocks and Sattler [51] ensures that each language induced by a knowledge base is regular, this restriction can be unnecessarily restrictive—even a knowledge base that does not contain complex role inclusions can violate it. While designing the OWL 2 specification, Motik et al. [78] generalised the regularity restriction by Horrocks and Sattler [51] to capture a wider spectrum of knowledge bases. Unfortunately, we show that the regularity restriction in the OWL 2 specification is incorrect because it does not ensure that each language $\mathcal{L}(R)$ induced by a knowledge base is regular. To address this problem, we propose a revised version of the syntactic condition on role inclusions for OWL 2 that generalises the condition by Horrocks and Sattler [51], correctly ensures that each language $\mathcal{L}(R)$ is regular, and captures each knowledge base that does not contain complex role inclusions. All the query answering algorithms that we present in this thesis support knowledge bases with complex role inclusions that satisfy our revised regularity condition.

In this thesis, we present in a unified framework a number of results that have already been published.

- A preliminary version of the CQ answering algorithm for $\mathcal{ELHO}_{\bot}^{+}$ without transitive and reflexive roles was presented at the 27th AAAI conference on Artificial Intelligence [95].

- The CQ answering algorithm for $\mathcal{ELHO}_{\bot}^{+}$, and the complexity results for acyclic CQs were presented at the 29th AAAI conference on Artificial Intelligence [94].

- The CQ answering algorithm for $\mathcal{ELRO}_{\bot}^{+}$, the encoding of the languages induced by regular role inclusions using PDAs, and the complexity results for graph XPath queries were presented in the Journal of Artificial Intelligence Research [96].

Chapter 2

Preliminary Notions

In this chapter, we recapitulate the basic definitions of finite and pushdown automata, context-free grammars, rules and queries; and, we present some non-standard complexity classes. In the rest of this thesis, we denote with $[i..j]$ the set containing each natural number $k \in \mathbb{N}$ such that $i \leq k \leq j$.

2.1 Automata and Language Theory

In the following, we use the standard notions of *alphabets*, *words*, *word concatenation*, *Kleene operators*, and *languages* from formal language theory [50]. We assume that alphabets do not contain the special symbol ε, which we use to label transitions in automata that do not consume input symbols. Furthermore, ϵ is the empty word. Finally, for w and w' words, $|w|$ is the number of symbols occurring in w, and $w - w'$ is the unique word $w'' \in \Sigma^*$ such that $w = w'' \cdot w'$ if such w'' exists, otherwise $w - w'$ is undefined.

2.1.1 Finite Automata

A *finite automaton* (FA) is a tuple $\mathcal{F} = \langle Q, \Sigma, \delta, i, f \rangle$ where Q is a finite set of states, Σ is a finite *input alphabet*, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ is a *transition function*, $i \in Q$ is the *start state*, and $f \in Q$ is the *final state*. Such $\mathcal{F}$ is *deterministic* if $|\delta(\lambda, \varepsilon)| = 0$ and $|\delta(\lambda, c)| \leq 1$ for each $\lambda \in Q$ and each $c \in \Sigma$; otherwise, $\mathcal{F}$ is *nondeterministic*.

An *instantaneous description* of $\mathcal{F}$ is a pair $\langle \lambda, w \rangle$ consisting of a state $\lambda \in Q$ and a word

$w \in \Sigma^*$. The *derivation relation* $\vdash$ *for* $\mathcal{F}$ is the smallest relation such that, for all states λ and λ' in Q, each symbol $c \in \Sigma$, and each word $w \in \Sigma^*$, we have that

- $\lambda' \in \delta(\lambda, c)$ implies that $\langle \lambda, c \cdot w \rangle \vdash \langle \lambda', w \rangle$; and
- $\lambda' \in \delta(\lambda, \varepsilon)$ implies that $\langle \lambda, w \rangle \vdash \langle \lambda', w \rangle$.

Let $\vdash^*$ be the reflexive-transitive closure of $\vdash$. Then, the *language accepted by* $\mathcal{F}$ is defined as $\mathcal{L}(\mathcal{F}) = \{w \in \Sigma^* \mid \langle i, w \rangle \vdash^* \langle f, \epsilon \rangle\}$. A language $\mathcal{L}$ is *regular* if and only if an FA $\mathcal{F}$ exists such that $\mathcal{L} = \mathcal{L}(\mathcal{F})$.

2.1.2 Context-Free Grammars and Pushdown Automata

A *context-free grammar* (CFG) is a tuple $G = \langle V, \Sigma, \Theta, S \rangle$ where V is a finite set of *non-terminal symbols*, Σ is a finite *input alphabet* such that $V \cap \Sigma = \emptyset$, Θ is a finite set of *production rules* of the form $T \to \gamma$ such that $T \in V$ and γ is a word over $(V \setminus \{S\}) \cup \Sigma$, and $S \in V$ is the *start symbol*.

The *derivation relation* $\Rightarrow$ *for* G is the smallest relation such that, for all words γ_1 and γ_2 in $(V \cup \Sigma)^*$ and each production rule $T \to \gamma \in \Theta$, we have $\gamma_1 \cdot T \cdot \gamma_2 \Rightarrow \gamma_1 \cdot \gamma \cdot \gamma_2$. Let $\Rightarrow^*$ be the reflexive-transitive closure of $\Rightarrow$. Then, $\mathcal{L}(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$ is the *language generated by* G. A language $\mathcal{L}$ is *context-free* if and only if a context-free grammar G exists such that $\mathcal{L} = \mathcal{L}(G)$.

A context-free grammar G is in *normal form* if each production rule in Θ is in one of the following forms: $T \to T_1 \cdot T_2$, $T \to c$, or $S \to \epsilon$, where T, T_1, and T_2 are non-terminal symbols of G, $c \in \Sigma$, and S is the start symbol of G. Each CFG can be normalised in polynomial time without affecting the language it generates [50].

A *pushdown automaton* (PDA) is a tuple $\mathcal{P} = \langle Q, \Sigma, \Gamma, \delta, i, I, f, F \rangle$ where Q is a finite set of *states*; Σ is a finite *input alphabet*; Γ is a finite *stack alphabet*; δ is a *transition function* mapping each state $\lambda \in Q$, each symbol $c \in (\Sigma \cup \{\varepsilon\})$, and each stack symbol $X \in \Gamma$ to a finite subset $\delta(\lambda, c, X) \subseteq Q \times \Gamma^*$; $i \in Q$ is the *start state*; $I \in \Gamma^*$ is the *start stack*; $f \in Q$ is the *final state*; and $F \in \Gamma^*$ is the *final stack*.

An *instantaneous description* of $\mathcal{P}$ is a triple $\langle \lambda, w, \gamma \rangle$ such that $\lambda \in Q$, $w \in \Sigma^*$, and $\gamma \in \Gamma^*$. We read the stack content γ from left to right; hence the leftmost symbol in γ is

the top of the stack. The *derivation relation* $\vdash$ *for* $\mathcal{P}$ is the smallest relation such that, for all states λ and λ' in Q, each symbol $c \in \Sigma$, each word $w \in \Sigma^*$, each stack symbol $X \in \Gamma$, and all words γ and γ' in Γ^*, we have

- $\langle \lambda', \gamma' \rangle \in \delta(\lambda, c, X)$ implies that $\langle \lambda, c \cdot w, X \cdot \gamma \rangle \vdash \langle \lambda', w, \gamma' \cdot \gamma \rangle$; and
- $\langle \lambda', \gamma' \rangle \in \delta(\lambda, \varepsilon, X)$ implies that $\langle \lambda, w, X \cdot \gamma \rangle \vdash \langle \lambda', w, \gamma' \cdot \gamma \rangle$.

Let $\vdash^*$ be the reflexive-transitive closure of relation $\vdash$. The *language accepted by* $\mathcal{P}$ is defined as $\mathcal{L}(\mathcal{P}) = \{w \in \Sigma^* \mid \langle i, w, I \rangle \vdash^* \langle f, \epsilon, F \rangle\}$.

Our definitions of a PDA $\mathcal{P}$ and of a language $\mathcal{L}(\mathcal{P})$ are somewhat non-standard. The literature typically considers a *Hopcroft PDA* [50] $\mathcal{P}_h$ that differs from our definition in that $\mathcal{P}_h$ does not contain the final stack F and the initial stack I of $\mathcal{P}_h$ is a symbol from Γ (rather than a word over Γ). Moreover, Hopcroft et al. [50] define the language accepted by $\mathcal{P}_h$ as $\mathcal{L}_h(\mathcal{P}_h) = \{w \in \Sigma^* \mid \exists \gamma \in \Gamma^* : \langle i, w, I \rangle \vdash^* \langle f, \epsilon, \gamma \rangle\}$. Hopcroft et al. [50] also show that we can construct in polynomial time a CFG G such that $\mathcal{L}_h(\mathcal{P}_h) = \mathcal{L}(G)$; thus Hopcroft PDAs accept precisely the context-free languages. Next, we show that our definitions are equivalent to the usual ones by Hopcroft et al. [50].

Proposition 2.1. *The following three properties hold.*

(1) For each PDA $\mathcal{P}$, one can compute in polynomial time a Hopcroft PDA $\mathcal{P}_h$ such that $\mathcal{L}(\mathcal{P}) = \mathcal{L}_h(\mathcal{P}_h)$.

(2) For each Hopcroft PDA $\mathcal{P}_h$, one can compute in polynomial time a PDA $\mathcal{P}$ such that $\mathcal{L}_h(\mathcal{P}_h) = \mathcal{L}(\mathcal{P})$.

(3) For each PDA $\mathcal{P}$, one can construct in polynomial time a normalised CFG G such that $\mathcal{L}(\mathcal{P}) = \mathcal{L}(G)$.

Proof (Sketch). Please note that property (3) follows immediately from property (1) and the results presented by Hopcroft et al. [50]; so in the following we first prove property (1), and then property (2).

Property (1). We show how to transform in polynomial time an arbitrary PDA $\mathcal{P}$ into a Hopcroft PDA $\mathcal{P}_h$ such that $\mathcal{L}(\mathcal{P}) = \mathcal{L}_h(\mathcal{P}_h)$. Such $\mathcal{P}_h$ uses a fresh initial state i' and

fresh stack symbols Z_0 and $\perp$ not occurring in Γ. Symbol Z_0 is the start stack symbol of $\mathcal{P}_h$; furthermore, $\mathcal{P}_h$ has a new ε-transition that moves the PDA from state i' to the initial state i of $\mathcal{P}$ by replacing Z_0 with $I \cdot \perp$, where I is the start stack of $\mathcal{P}$. At this point, $\mathcal{P}_h$ simulates $\mathcal{P}$, always leaving $\perp$ at the bottom of the stack until it reaches the final state f of $\mathcal{P}$. Next, $\mathcal{P}_h$ uses fresh states $\lambda_1, \ldots, \lambda_{|F|}$ and fresh ε-transitions that move $\mathcal{P}_h$ from state f to $\lambda_{|F|}$ by reading F from the stack. Finally, from $\lambda_{|F|}$, PDA $\mathcal{P}_h$ ε-moves to a fresh final state f' if the top-most symbol on the stack is $\perp$, thus accepting the input whenever $\mathcal{P}$ reaches f with F on its stack. Automata $\mathcal{P}$ and $\mathcal{P}_h$ clearly accept the same languages.

Property (2). We show how to transform in polynomial time an arbitrary Hopcroft PDA $\mathcal{P}_h$ into a PDA $\mathcal{P}$ such that $\mathcal{L}_h(\mathcal{P}_h) = \mathcal{L}(\mathcal{P})$. PDA $\mathcal{P}$ uses a fresh stack symbol $\perp$, its initial stack is $I \cdot \perp$ where I is the initial stack symbol of $\mathcal{P}_h$, and its final stack is the empty word. Then $\mathcal{P}$ simulates $\mathcal{P}_h$, always leaving $\perp$ at the bottom of the stack until it reaches the final state f of $\mathcal{P}_h$. Next, $\mathcal{P}$ ε-moves to a fresh final state f' and pops the topmost symbol off the stack. Finally, $\mathcal{P}$ takes further ε-transitions to empty its stack, eventually reaching its final state with the empty stack. Automata $\mathcal{P}$ and $\mathcal{P}_h$ clearly accept the same languages. □

For k a natural number, the *k-bounded language accepted by* $\mathcal{P}$ is the set $\mathcal{L}_k(\mathcal{P})$ containing each word $w \in \Sigma^*$ for which a derivation $\langle \lambda_0, w_0, \gamma_0 \rangle \vdash \cdots \vdash \langle \lambda_n, w_n, \gamma_n \rangle$ exists where

- λ_0 and λ_n are the start and the final state of $\mathcal{P}$, respectively,
- $w_0 = w$ and $w_n = \epsilon$,
- γ_0 and γ_n are the start and the final stack of $\mathcal{P}$, respectively, and
- $|\gamma_i| \leq k$ for each $i \in [0..n]$.

Then, $\mathcal{P}$ has a *k-bounded stack* if $\mathcal{L}(\mathcal{P}) = \mathcal{L}_k(\mathcal{P})$. As the stack of $\mathcal{P}$ is bounded by a constant, PDA $\mathcal{P}$ can be simulated by an FA that encodes the stack contents using its states, and so $\mathcal{L}(\mathcal{P})$ is regular, but translating $\mathcal{P}$ into an FA may require space exponential in k [34]. In contrast, the following proposition shows that a PDA $\mathcal{P}_k$ exists such that $\mathcal{L}(\mathcal{P}_k) = \mathcal{L}_k(\mathcal{P})$ and the size of $\mathcal{P}_k$ is polynomial in the size of $\mathcal{P}$ and k.

Proposition 2.2. *For each PDA $\mathcal{P}$ and each natural number $k \in \mathbb{N}$, one can compute in polynomial time a PDA $\mathcal{P}_k$ such that $\mathcal{L}(\mathcal{P}_k) = \mathcal{L}_k(\mathcal{P})$.*

Proof. Let $\mathcal{P} = \langle Q, \Sigma, \Gamma, \delta, i, I, f, F \rangle$ be a PDA and let $k \in \mathbb{N}$ be a natural number. Let $\mathcal{P}_k = \langle Q_k, \Sigma, \Gamma, \delta_k, i_k, I, f_k, F \rangle$ where set Q_k, relation δ_k, and states i_k and f_k are as follows.

- $Q_k = Q \times [0..k]$.
- δ_k is the smallest transition function such that, for each $\ell \in [0..k]$, each symbol $c \in \{\varepsilon\} \cup \Sigma$, all states $\lambda, \lambda' \in Q$, and each word $\gamma \in \Gamma^*$ with $\langle \lambda', \gamma \rangle \in \delta(\lambda, c, X)$ and $\ell + |\gamma| - 1 \leq k$, we have that $\langle \langle \lambda', \ell + |\gamma| - 1 \rangle, \gamma \rangle \in \delta_k(\langle \lambda, \ell \rangle, c, X)$.
- $i_k = \langle i, |I| \rangle$ and $f_k = \langle f, |F| \rangle$.

Clearly, $\mathcal{P}_k$ can be computed in time polynomial in the size of $\mathcal{P}$ and k. Let $\vdash$ and $\vdash_k$ be the derivation relations for $\mathcal{P}$ and $\mathcal{P}_k$, respectively. By the definitions of δ_k and i_k, we have that $\langle \langle \lambda, \ell \rangle, w, \gamma \rangle \vdash_k \langle \langle \lambda', j \rangle, w', \gamma' \rangle$ if and only if $\langle \lambda, w, \gamma \rangle \vdash \langle \lambda', w', \gamma' \rangle$, $|\gamma| = \ell$ and $|\gamma'| = j$, and $\max(\ell, j) \leq k$. Thus, we have $\mathcal{L}_k(\mathcal{P}) = \mathcal{L}(\mathcal{P}_k)$, as required. □

2.2 Rules and Queries

In this section, we start by recapitulating the definition of function-free first-order logic with equality, which provides us with a logical framework that encapsulates both rules and queries. Then, we present the notions of first-order rule bases, and instance and conjunctive queries. Finally, we introduce a special version of the oblivious chase [2, 74, 12], and we describe the so-called universal interpretation that our chase variant produces for a given rule base. In the following, we often identify a conjunction with the set of its conjuncts.

2.2.1 First-Order Logic with Equality

A *first-order signature* consists of countably infinite and mutually disjoint sets Φ_P, Φ_C, and Φ_V of *predicates*, *constants*, and *variables*, respectively. Each predicate is associated with a nonnegative *arity* $n \in \mathbb{N}$. We assume that Φ_P does not contain the *equality* symbol $\approx$.

A *term* is a constant or a variable. An *equality* is an expression of the form $t_1 \approx t_2$ where t_1 and t_2 are terms. An *atom* is an expression of the form $P(t_1, \ldots, t_n)$ where P is an n-ary predicate and each t_i is a term. An *atomic formula* is either an equality or an atom. For α a term or an atomic formula, α is *ground* if it does not contain variables.

The set F of *first-order formulas* is the smallest set that contains each atomic formula, and such that the following three properties hold.

- For each formula $\varphi \in F$, set F contains $\neg\varphi$ (negation).
- For all formulas φ and ψ in F, set F contains $\varphi \wedge \psi$ (conjunction), $\varphi \vee \psi$ (disjunction), and $\varphi \to \psi$ (implication).
- For each formula $\varphi \in F$ and each variable $x \in \Phi_V$, set F contains $\forall x.\varphi$ (universal quantification) and $\exists x.\varphi$ (existential quantification).

A variable x in a formula φ is *free* if it does not occur within the scope of a quantifier. We sometimes write $\varphi(\vec{x})$ to emphasise that the free variables of φ are precisely those listed in $\vec{x}$. A formula that does not contain free variables is a *sentence*. A *first-order theory* is a finite set of sentences.

The semantics of first-order logic with equality is defined in terms of interpretations. A *first-order interpretation* is a pair $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ where $\Delta^{\mathcal{I}}$ is a non-empty set of domain elements, called the *domain of* $\mathcal{I}$, and $\cdot^{\mathcal{I}}$ is an *interpretation function* that maps each constant $c \in \Phi_C$ to an element $c^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each predicate $P \in \Phi_P$ of arity n to a relation $P^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^n$, and the equality symbol $\approx$ to the identity relation $\approx^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$. According to this definition, the interpretation function can map distinct constants c and d to the same domain element—that is, we do not make *unique name assumption*.

An *assignment* ν is a mapping of variables to elements in $\Delta^{\mathcal{I}}$; for convenience, we extend each assignment ν by setting $\nu(c) = c^{\mathcal{I}}$ for each constant $c \in \Phi_C$. For a variable x, assignment ν_x is an *x-variant* of ν if $\nu_x(y) = \nu(y)$ for each variable $y \in \mathsf{dom}(\nu)$ with $y \neq x$. The notion of *satisfaction* of a formula φ in a first-order interpretation $\mathcal{I}$ w.r.t. an assignment ν, written $\mathcal{I} \models_\nu \varphi$, is defined inductively as follows.

- $\mathcal{I} \models_\nu P(t_1, \ldots, t_n)$ if and only if $\langle \nu(t_1), \ldots, \nu(t_n) \rangle \in P^{\mathcal{I}}$.
- $\mathcal{I} \models_\nu t_1 \approx t_2$ if and only if $\nu(t_1) = \nu(t_2)$.
- $\mathcal{I} \models_\nu \neg\varphi$ if and only if $\mathcal{I} \not\models_\nu \varphi$.
- $\mathcal{I} \models_\nu \varphi \wedge \psi$ if and only if $\mathcal{I} \models_\nu \varphi$ and $\mathcal{I} \models_\nu \psi$.

- $\mathcal{I} \models_\nu \varphi \vee \psi$ if and only if $\mathcal{I} \models_\nu \varphi$ or $\mathcal{I} \models_\nu \psi$.
- $\mathcal{I} \models_\nu \varphi \to \psi$ if and only if $\mathcal{I} \not\models_\nu \varphi$ or $\mathcal{I} \models_\nu \psi$.
- $\mathcal{I} \models_\nu \exists x.\varphi$ if and only if an x-variant ν_x of ν exists such that $\mathcal{I} \models_{\nu_x} \varphi$.
- $\mathcal{I} \models_\nu \forall x.\varphi$ if and only if, for each x-variant ν_x of ν, we have that $\mathcal{I} \models_{\nu_x} \varphi$.

When φ is a sentence, the satisfaction of φ in $\mathcal{I}$ does not depend on the choice of the assignment ν, so we simply write $\mathcal{I} \models \varphi$.

An interpretation $\mathcal{I}$ is a *first-order model* of a theory T if $\mathcal{I} \models \varphi$ for each sentence $\varphi \in T$. A theory T is *first-order consistent* if a first-order model of T exists, otherwise T is *first-order inconsistent*.

2.2.2 First-Order Rules

In this thesis, we consider two types of rules. An *existential rule* is a formula of the form

$$\forall \vec{x} \forall \vec{y}.\, \varphi(\vec{x}, \vec{y}) \to \exists \vec{z}.\, \psi(\vec{x}, \vec{z}) \tag{2.1}$$

where φ and ψ are non-empty conjunctions of atoms. An *equality rule* is a formula of the form

$$\forall \vec{x}.\, \varphi(\vec{x}, \vec{y}) \to s \approx t \tag{2.2}$$

where φ is a non-empty conjunction of atoms and s and t are terms with variables in $\vec{x}$. In the rest of this thesis, we typically omit writing universal quantifiers in front of rules.

A *rule base* Σ is a first-order theory that consists only of rules and ground atoms. Such a rule base Σ is a *datalog program* if $\vec{z} = \emptyset$ for each rule of the form (2.1) occurring in Σ.

Note that Σ is always consistent as we do not make the unique name assumption, rules do not contain negation, and our definition of first-order logic does not include the primitives $\top$ and $\bot$ representing true and false, respectively. Finally, the *set of consequences of* Σ contains each ground atomic formula φ such that $\mathcal{I} \models \varphi$ for each first-order model $\mathcal{I}$ of Σ.

2.2.3 Instance and Conjunctive Queries

A *conjunctive query* (CQ) is a formula $q = \exists \vec{y}.\, \psi(\vec{x}, \vec{y})$ where ψ is a non-empty conjunction of atoms. The free variables $\vec{x}$ in q are the *answer variables* of q. Let $\mathsf{var}(q) = \vec{x} \cup \vec{y}$ and let $\mathsf{term}(q)$ be the set of all terms occurring in q. When $\vec{x}$ is empty, we call $q = \exists \vec{y}.\, \psi(\vec{y})$ a *Boolean CQ*. An *instance query* is a CQ of the form $q = A(x)$ where x is a variable and A is a unary predicate.

A *substitution* τ is a partial mapping of variables to terms; $\mathsf{dom}(\tau)$ and $\mathsf{rng}(\tau)$ are the domain and the range of τ, respectively. Moreover, $\tau|_V$ is the restriction of τ to a set of variables $V \subseteq \Phi_V$. For convenience, we extend each substitution τ by setting $\tau(c) = c$ for each constant $c \in \Phi_C$.

Let τ be a substitution and let $q = \exists \vec{y}.\, \psi(\vec{x}, \vec{y})$ be a CQ. The *result* of applying τ to q is the CQ $\tau(q) = \exists \vec{y}_\tau.\, \psi_\tau$ where $\vec{y}_\tau$ and ψ_τ are defined as follows.

- $\vec{y}_\tau$ is obtained from $\vec{y}$ by first removing each variable $y \in \vec{y}$ such that $\tau(y)$ is a ground term, and then by replacing each remaining variable y with $\tau(y) \in \Phi_V$.
- ψ_τ is the result of simultaneously replacing each variable x occurring in ψ with $\tau(x)$.

When $\mathsf{dom}(\tau) = \vec{x}$ and each element of $\mathsf{rng}(\tau)$ is a constant, $\tau(q)$ is the Boolean CQ obtained by replacing each variable $x \in \vec{x}$ with $\tau(x) \in \Phi_C$. Example 2.1 shows that our definition of $\tau(q)$ is somewhat non-standard because variables that are not free in q can also be replaced.

Example 2.1. *Consider the following CQ q and substitution τ.*

$$q = \exists y_1 \exists y_2 \exists y_3.\, R(y_1, y_2) \wedge T(y_1, y_3) \tag{2.3}$$

$$\tau = \{y_2 \mapsto a, y_3 \mapsto y_4\} \tag{2.4}$$

Then we have $\tau(q) = \exists y_1 \exists y_4.\, R(y_1, a) \wedge T(y_1, y_4)$.

Let Σ be a rule base and let $q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$ be a CQ. A substitution π is a *certain answer* to q over Σ, written $\Sigma \models \pi(q)$, if $\mathsf{dom}(\pi) = \vec{x}$, each element in $\mathsf{rng}(\pi)$ is a constant occurring in Σ, and $\mathcal{I} \models \pi(q)$ for each first-order model $\mathcal{I}$ of Σ. When q is a Boolean CQ, π being a certain answer to q over Σ implies that $\pi = \emptyset$ and $\pi(q) = q$; so we simply write $\Sigma \models q$.

2.2.4 Chase and Universal Interpretations

We present our version of the oblivious chase and describe the so-called universal interpretation of a rule base Σ that our chase produces. For the rest of this section, we fix an arbitrary rule base Σ.

Our chase variant is based on a countably infinite set Φ_N of *labelled nulls* pairwise disjoint with Φ_C, Φ_P, and Φ_V, and on a total order $<$ on $\Phi_C \cup \Phi_N$ such that $c < w$ for each constant $c \in \Phi_C$ and each labelled null $w \in \Phi_N$. In the rest of this thesis, we extend the notion of terms to allow for labelled nulls. Hence, a term is a constant, a variable, or a labelled null; a ground term is a constant or a labelled null. Furthermore, we extend each substitution τ by setting $\tau(w) = w$ for each labelled null $w \in \Phi_N$, and we call τ *ground* if $\mathsf{rng}(\tau)$ contains only ground terms. This allows atoms to contain labelled nulls, which we will use to represent assertions whose existence is implied by existential rules in Σ.

Let $\rightsquigarrow$ be a fresh binary predicate not occurring in Σ. An *instance* is a set $I = \mathsf{inst}_I \cup \mathsf{eq}_I$ where inst_I is a set of atoms over the predicates that occur in Σ, and eq_I is a set of atoms of the form $w \rightsquigarrow w'$. A ground term w is *irreducible* in I if, for each term w', we have that $w \rightsquigarrow w' \notin I$. Relation $\rightsquigarrow_I^*$ is the smallest reflexive and transitive relation on ground terms such that $w \rightsquigarrow_I^* w'$ for each $w \rightsquigarrow w' \in I$. For each ground term w, let $\|w\|_I = w'$ where w' is the $<$-smallest irreducible term occurring in I such that $w \rightsquigarrow_I^* w'$. For a formula φ, $\|\varphi\|_I$ is obtained by replacing each ground term w occurring in φ with $\|w\|_I$. For a sequence $\rho = R_1 \cdots R_m$ of binary predicates and terms w and w', we write $\rho(w, w') \in \mathsf{inst}_I$ if terms $w = w_0, \ldots, w_m = w'$ exist such that $R_i(w_{i-1}, w_i) \in \mathsf{inst}_I$ for each $i \in [1..m]$.

We next present the notion of a *trigger* [81] for an instance I and, in analogy with rules, we define both existential and equality triggers. Let I be an instance. An *existential trigger* for I is a pair $\langle r, \sigma \rangle$ where $r \in \Sigma$ is an existential rule of the form $r = \varphi(\vec{x}, \vec{y}) \rightarrow \exists \vec{z}.\psi(\vec{x}, \vec{z})$ and σ is a ground substitution with $\mathsf{dom}(\sigma) = \vec{x} \cup \vec{y}$ such that $\sigma(\|\varphi\|_I) \subseteq \mathsf{inst}_I$. The *result* of applying an existential trigger $\langle r, \sigma \rangle$ to I is the instance J where

- κ is the substitution that maps each variable $x \in \vec{x}$ according to σ, and each variable $z \in \vec{z}$ to the smallest, fresh labelled null $\kappa(z) \in \Phi_N$ that is $<$-larger than all terms occurring in I,

- $\mathsf{inst}_J = \mathsf{inst}_I \cup \kappa(\|\psi\|_I)$, and
- $\mathsf{eq}_J = \mathsf{eq}_I$.

Each fresh labelled null occurring in J is said to be *generated by* rule r. An *equality trigger* for I is a pair $\langle r, \sigma \rangle$ where $r \in \Sigma$ is an equality rule of the form $r = \varphi(\vec{x}, \vec{y}) \rightarrow s \approx t$ and σ is a ground substitution with $\mathsf{dom}(\sigma) = \vec{x} \cup \vec{y}$ such that $\sigma(\|\varphi\|_I) \subseteq \mathsf{inst}_I$ and $\|\sigma(s)\|_I \neq \|\sigma(t)\|_I$. The *result* of applying an equality trigger $\langle r, \sigma \rangle$ to I is the instance J where

- w and w' are the $<$-largest and $<$-smallest terms in $\{\|\sigma(s)\|_I, \|\sigma(t)\|_I\}$, respectively,
- inst_J is obtained from I by replacing each occurrence of term w with w', and
- $\mathsf{eq}_J = \mathsf{eq}_I \cup \{w \rightsquigarrow w'\}$.

A *chase sequence for* Σ *w.r.t.* $<$ is a sequence of triples $\langle I_1, r_1, \sigma_1 \rangle, \langle I_2, r_2, \sigma_2 \rangle, \ldots$ indexed by a (possibly empty) set $\mathbb{S} = \{1, 2, \ldots\}$ of positive natural numbers, where I_0 is the instance containing each ground atom in Σ and, for each $i \in \mathbb{S}$, the following hold:

- $\langle r_i, \sigma_i \rangle$ is a trigger for I_{i-1} and I_i is the result of $\langle r_i, \sigma_i \rangle$ on I_{i-1}, and
- no index $j \in \mathbb{S}$ exists such that $j < i$ and $\langle r_j, \sigma_j \rangle = \langle r_i, \sigma_i \rangle$.

This sequence must be *fair*: for each $i \in \mathbb{S}$ and each trigger $\langle r, \sigma \rangle$ for I_{i-1}, either an index $j \in \mathbb{S}$ exists such that $\langle r_j, \sigma_j \rangle = \langle r, \sigma \rangle$, or an index $k \in \mathbb{S}$ with $k \geq i$ exists such that $\langle r, \sigma \rangle$ is not a trigger for I_k. Instance $I = \bigcup_{i \in \mathbb{S}} \bigcap_{j \geq i} I_j$ is a *universal interpretation* of Σ.

Our chase variant differs from the standard notion of oblivious chase [12, 81] in that we do not make the unique name assumption. Hence, the application of equality rules can replace constants, as well as labelled nulls, and our chase procedure never fails.

Let I be a universal interpretation of Σ. Set inst_I contains the atoms required to satisfy the rules in Σ; in contrast, set eq_I is 'auxiliary' and we use it to keep record on how terms have been identified during the chase sequence producing I. Please note that, for each term w occurring in inst_I, we have that $\|w\|_I = w$. Also, owing to the fairness of the chase sequence, instance I is *closed under* Σ—that is, for each trigger $\langle r, \sigma \rangle$ for I, an index $i \geq 0$ exists such that I_{i+1} is the result of applying $\langle r, \sigma \rangle$ to I_i. Furthermore, for each universal

interpretation J of Σ, we have that inst_I is homomorphically equivalent to inst_J and, if Σ is a datalog program, then inst_I equals inst_J; however, I and J may disagree on the atoms over predicate $\rightsquigarrow$ [81]. Finally, it is well known [1, 31, 12] that inst_I can be homomorphically embedded into any first-order model of Σ; hence, I can be used to answer CQs over Σ.

Theorem 2.3 (Adapted from [12]). *For each CQ $q = \exists \vec{y}.\, \psi(\vec{x}, \vec{y})$, and each substitution π, $\Sigma \models \pi(q)$ if and only if a ground substitution π_* with $\mathsf{dom}(\pi_*) = \vec{x} \cup \vec{y}$ exists such that $\pi = \pi_*|_{\vec{x}}$, each element in $\mathsf{rng}(\pi_*|_{\vec{x}})$ is a constant occurring in Σ, and $\|\pi_*(q)\|_I \subseteq \mathsf{inst}_I$.*

2.3 Complexity Theory

In this thesis, we provide several complexity results for decision problems related to answering conjunctive queries over a specific form of logical theories, called description logic knowledge bases, which we formally introduce in the next chapter. We next recapitulate several relevant notions from the theory of computational complexity; for more details, we refer the reader to the textbook by Papadimitriou [84].

The complexity of a decision problem is typically measured in terms of the resources, time or space, a Turing machine (TM) uses to solve the problem w.r.t. the size of the input. For α a conjunctive query or a knowledge base, the *size* $|\alpha|$ of α is the number of symbols that is required to write α on the tape of a TM whose input alphabet Σ contains the binary digits 0 and 1 and all logical symbols occurring in α, but Σ's size is independent of α. To this end, we assume a suitable binary encoding of predicates, constants, and variables such as the encoding by Abiteboul et al. [1].

Complexity classes group decision problems based on the amount of resources a TM uses to solve the problem. The class NL contains each decision problem that can be solved by a nondeterministic TM using logarithmic space. The class PTime (resp. NP) contains each decision problem that can be solved on a deterministic (resp. nondeterministic) TM using polynomial time. The class PSpace (resp. NPSpace) contains each decision problem that can be solved on a deterministic (resp., nondeterministic) TM using polynomial space; Savitch [90] showed that PSpace equals NPSpace.

In the following, we will infer complexity bounds using log-space many-one reductions.

Chapter 3

The EL Profile of OWL 2

OWL 2 EL is one of the three profiles of OWL 2 and is based on the $\mathcal{EL}$ family of DLs [4]. We next present the DL $\mathcal{ELRO}^{+}_{\bot}$ that logically underpins OWL 2 EL, as well as other members of the $\mathcal{EL}$ family of DLs, and formally introduce the problem of answering CQs over knowledge bases. In the rest of this chapter, we fix a first-order signature consisting of sets Φ_P, Φ_C, and Φ_V of unary and binary predicates, constants, and variables, respectively.

3.1 The $\mathcal{EL}$ Family of DLs

We start by introducing the DL $\mathcal{ELRO}^{+}_{\bot}$ that logically underpins OWL 2 EL, and subsequently we specialise the definition of $\mathcal{ELRO}^{+}_{\bot}$ to obtain the other members of the family.

3.1.1 The DL $\mathcal{ELRO}^{+}_{\bot}$

Our definition of $\mathcal{ELRO}^{+}_{\bot}$ is based on three countably infinite and mutually disjoint sets $N_{\mathfrak{C}} \subseteq \Phi_P$, $N_{\mathfrak{R}} \subseteq \Phi_P$, and $N_{\mathfrak{I}} \subseteq \Phi_C$ of unary predicates, binary predicates, and constants. In DL parlance, unary predicates are called *atomic concepts*, binary predicates are called *roles*, and constants are called *individuals*. We assume that $\{\top_c, \bot_c\} \subseteq N_{\mathfrak{C}}$, where $\top_c$ is the *top concept* and $\bot_c$ is the *bottom concept*; similarly, we assume that $\{\top_r, \bot_r\} \subseteq N_{\mathfrak{R}}$, where $\top_r$ is the *top role* (universal role) and $\bot_r$ is the *bottom role*.

A *role chain* ρ is a word over the alphabet of roles; for $|\rho| = 0$, we call ρ the *empty role chain* and we write it as ϵ. The set $\mathcal{C}$ of $\mathcal{ELRO}^{+}_{\bot}$ *concepts* is the smallest set that contains

Table 3.1: Interpreting $\mathcal{ELRO}^+_\bot$ concepts, role chains, and axioms in an interpretation $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$

	Syntax	Semantics
Concepts:		
atomic concept	A	$A^\mathcal{I} \subseteq \Delta^\mathcal{I}$
top concept	$\top_c$	$\Delta^\mathcal{I}$
bottom concept	$\bot_c$	$\emptyset$
nominal	$\{a\}$	$\{a^\mathcal{I}\}$
conjunction	$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
self-restriction	$\exists R.\mathsf{Self}$	$\{x \in \Delta^\mathcal{I} \mid \langle x, x \rangle \in R^\mathcal{I}\}$
existential restriction	$\exists R.C$	$\{x \in \Delta^\mathcal{I} \mid \exists y \in C^\mathcal{I} : \langle x, y \rangle \in R^\mathcal{I}\}$
Role chains:		
role	R	$R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$
top role	$\top_r$	$\Delta^\mathcal{I} \times \Delta^\mathcal{I}$
bottom role	$\bot_r$	$\emptyset$
empty role chain	ϵ	$\{\langle x, x \rangle \mid x \in \Delta^\mathcal{I}\}$
non-empty role chain	$R_1 \cdots R_n$	$R_1^\mathcal{I} \circ \cdots \circ R_n^\mathcal{I}$
TBox axioms:		
concept inclusion	$C \sqsubseteq D$	$C^\mathcal{I} \subseteq D^\mathcal{I}$
range restriction	$\mathsf{range}(R, C)$	$R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times C^\mathcal{I}$
key	$\mathsf{key}(C, R_1 \ldots R_n)$	For all $x, y, z_1, \ldots, z_n$ in $\Delta^\mathcal{I}$ such that individuals $a, b, c_1, \ldots, c_n$ in $N_\mathfrak{I}$ exist with $x = a^\mathcal{I}$, $y = b^\mathcal{I}$, and $z_i = c_i^\mathcal{I}$ for $1 \leq i \leq n$, $x = y$ holds whenever $\{x, y\} \subseteq C^\mathcal{I}$ and $\{\langle x, z_i \rangle, \langle y, z_i \rangle\} \subseteq R_i^\mathcal{I}$ for $1 \leq i \leq n$.
RBox axioms:		
role inclusion	$\rho \sqsubseteq R$	$\rho^\mathcal{I} \subseteq R^\mathcal{I}$
ABox axioms:		
concept assertion	$A(a)$	$a^\mathcal{I} \in A^\mathcal{I}$
role assertion	$R(a, b)$	$\langle a^\mathcal{I}, b^\mathcal{I} \rangle \in \mathcal{R}^\mathcal{I}$

each atomic concept $A \in N_{\mathfrak{C}}$ such that the following four properties hold.

- For each individual $a \in N_{\mathfrak{I}}$, set $\mathcal{C}$ contains $\{a\}$ (*nominal*).
- For all concepts C and D in $\mathcal{C}$, set $\mathcal{C}$ contains $C \sqcap D$ (*conjunction*).
- For each role $R \in N_{\mathfrak{R}}$, set $\mathcal{C}$ contains $\exists R.\mathsf{Self}$ (*self-restriction*).
- For each role $R \in N_{\mathfrak{R}}$ and each concept $C \in \mathcal{C}$, set $\mathcal{C}$ contains $\exists R.C$ (*existential restriction*).

We next define the axioms that can occur in $\mathcal{ELRO}^{+}_{\bot}$ KBs and, as usual, we distinguish between assertional (ABox), terminological (TBox), and relational (RBox) axioms.

An *assertion* is an expression of the form $A(a)$ or $R(a, b)$ where A is an atomic concept, R is a role, and a and b are individuals; an *ABox* $\mathcal{A}$ is a finite set of assertions. Furthermore, a *concept inclusion* is an expression of the form $C \sqsubseteq D$ where C and D are concepts; a *range restriction* is an expression of the form $\mathsf{range}(R, C)$ where R is a role and C is a concept; a *key* is an expression of the form $\mathsf{key}(C, R_1, \ldots, R_n)$ where C is a concept and each R_i is a role; and a *TBox* $\mathcal{T}$ is a finite set of concept inclusions, range restrictions, and keys. Finally, a *role inclusion* is an expression $\rho \sqsubseteq R$ where ρ is a role chain and R is a role. An *RBox* $\mathcal{R}$ is a finite set of role inclusions; such $\mathcal{R}$ is a *role hierarchy* if, for each role inclusion $\rho \sqsubseteq R$ in $\mathcal{R}$, we have that $|\rho| \leq 1$ or $\rho = R \cdot R$.

For a set of axioms Φ, we define

$$\begin{aligned} \mathsf{con}_\Phi &= \{\top_c, \bot_c\} \cup \{A \in N_{\mathfrak{C}} \mid A \text{ occurs in } \Phi\}, \\ \mathsf{rol}_\Phi &= \{\top_r, \bot_r\} \cup \{R \in N_{\mathfrak{R}} \mid R \text{ occurs in } \Phi\}, \text{ and} \\ \mathsf{ind}_\Phi &= \{a \in N_{\mathfrak{I}} \mid a \text{ occurs in } \Phi\} \end{aligned}$$

Let $\mathcal{R}$ be an RBox. For a role $R \in \mathsf{rol}_\mathcal{R}$, R is *transitive* in $\mathcal{R}$ if $R \cdot R \sqsubseteq R \in \mathcal{R}$, and R is *reflexive* in $\mathcal{R}$ if $\epsilon \sqsubseteq R \in \mathcal{R}$. A role R is *composite* in $\mathcal{R}$ if a role inclusion $\rho \sqsubseteq R \in \mathcal{R}$ exists such that $|\rho| > 1$. The *subrole relation* $\sqsubseteq^*_\mathcal{R}$ for $\mathcal{R}$ is the smallest reflexive and transitive relation on $\mathsf{rol}_\mathcal{R}$ such that

- $S \sqsubseteq^*_\mathcal{R} R$ for each role inclusion $S \sqsubseteq R \in \mathcal{R}$, and

- $S \sqsubseteq^*_{\mathcal{R}} \top_r$ for each role $S \in \mathsf{rol}_{\mathcal{R}}$.

A role R is *simple* in $\mathcal{R}$ if, for each role $S \in \mathsf{rol}_{\mathcal{R}}$ with $S \sqsubseteq^*_{\mathcal{R}} R$, we have that $S \notin \{\top_r, \bot_r\}$, and S is not composite in $\mathcal{R}$.

An $\mathcal{ELRO}^+_\bot$ *knowledge base* is a finite set $\mathcal{K} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$ where $\mathcal{T}$ is a TBox, $\mathcal{R}$ is an RBox, and $\mathcal{A}$ is an ABox such that the following holds.

- For each concept $\exists R.\mathsf{Self}$ occurring in $\mathcal{T}$, role R is simple in $\mathcal{R}$.
- For each $R_1 \cdots R_n \sqsubseteq R \in \mathcal{R}$ with $n \geq 1$ and each $\mathsf{range}(R', C) \in \mathcal{T}$ with $R \sqsubseteq^*_{\mathcal{R}} R'$, a role R'_n exists such that $R_n \sqsubseteq^*_{\mathcal{R}} R'_n$ and $\mathsf{range}(R'_n, C) \in \mathcal{T}$.

The semantics of $\mathcal{ELRO}^+_\bot$ is defined using interpretations. A *description logic interpretation* is a pair $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ where $\Delta^{\mathcal{I}}$ is a non-empty set of domain elements, the *domain* of $\mathcal{I}$, and $\cdot^{\mathcal{I}}$ is the *interpretation function* that maps each individual $a \in N_{\mathfrak{I}}$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each atomic concept $A \in N_{\mathfrak{C}} \setminus \{\top_c, \bot_c\}$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each role $R \in N_{\mathfrak{R}} \setminus \{\top_r, \bot_r\}$ to a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Function $\cdot^{\mathcal{I}}$ is extended to role chains and concepts as shown in the upper part of Table 3.1, where $A \in N_{\mathfrak{C}}$, $R_{(i)} \in N_{\mathfrak{R}}$, $a \in N_{\mathfrak{I}}$, C and D are concepts, and $\circ$ denotes composition of binary relations.

Although DL interpretations are closely related to first-order interpretations as defined in Section 2.2, they differ in one key respect. Each DL interpretation interprets $\top_c$ and $\bot_c$, and $\top_r$ and $\bot_r$ as shown in Table 3.1; in contrast, in our definition of first-order logic $\top_c$ and $\bot_c$, and $\top_r$ and $\bot_r$ are just standard predicates without any predetermined meaning. In other words, first-order interpretations do not ascribe a predetermined meaning to $\top_c$ and $\bot_c$, and $\top_r$ and $\bot_r$. Hence, each DL interpretation is a first-order interpretation, but the converse does not necessarily hold.

A DL interpretation $\mathcal{I}$ is a *description logic model* of $\mathcal{K}$ if $\mathcal{I}$ satisfies all axioms occurring in $\mathcal{K}$ as shown at the bottom of Table 3.1. Moreover, $\mathcal{K}$ is *consistent* if a DL model of $\mathcal{K}$ exists, otherwise $\mathcal{K}$ is *inconsistent*. Knowledge base consistency can be decided in time polynomial in $|\mathcal{K}|$ [4, 64].

Finally, we associate each role R occurring in an RBox $\mathcal{R}$ with the language $\mathcal{L}(R)$ induced by the role inclusions in $\mathcal{R}$. The *rewrite relation* $\Rightarrow$ *for* $\mathcal{R}$ is the smallest relation on role chains such that the following holds for all role chains ρ_1 and ρ_2.

- $\rho_1 \cdot R \cdot \rho_2 \Rightarrow \rho_1 \cdot \rho \cdot \rho_2$ for each axiom $\rho \sqsubseteq R \in \mathcal{R}$.
- $\rho_1 \cdot \top_r \cdot \rho_2 \Rightarrow \rho_1 \cdot \rho \cdot \rho_2$ for each role chain $\rho \in (\mathsf{rol}_{\mathcal{R}})^*$.

Then, relation $\Rightarrow^*$ is the reflexive and transitive closure of $\Rightarrow$ and, for each role $R \in \mathsf{rol}_{\mathcal{R}}$, $\mathcal{L}(R) = \{\rho \in (\mathsf{rol}_{\mathcal{R}})^* \mid R \Rightarrow^* \rho\}$ is the *language induced by RBox* $\mathcal{R}$. When $\mathcal{R}$ is a role hierarchy, for all roles S and R in $\mathsf{rol}_{\mathcal{R}}$, we have $S \in \mathcal{L}(R)$ if and only if $S \sqsubseteq^*_{\mathcal{R}} R$.

3.1.2 The Other Members of the Family

Other important members of the $\mathcal{EL}$ family of DLs can be characterised by restricting the shape of an $\mathcal{ELRO}^+_\bot$ knowledge base $\mathcal{K} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$ as follows.

- $\mathcal{K}$ is in $\mathcal{ELHO}^+_\bot$ if $\mathcal{R}$ is a role hierarchy.
- $\mathcal{K}$ is in $\mathcal{ELHO}_\bot$ if $\mathcal{K}$ is in $\mathcal{ELHO}^+_\bot$ and $\mathcal{K}$ does not contain the top and bottom role, the TBox $\mathcal{T}$ does not contain keys and concept inclusions involving self-restrictions, and each role inclusion $\rho \sqsubseteq R \in \mathcal{R}$ is such that $|\rho| = 1$.
- $\mathcal{K}$ is in $\mathcal{ELH}_\bot$ if $\mathcal{K}$ is in $\mathcal{ELHO}_\bot$ and the TBox $\mathcal{T}$ does not contain range restriction and concept inclusions involving nominals.
- $\mathcal{K}$ is in $\mathcal{EL}$ if $\mathcal{K}$ is in $\mathcal{ELH}_\bot$ and $\mathcal{K}$ does not contain the bottom concept, and $\mathcal{R} = \emptyset$.

3.2 CQ Answering in the $\mathcal{EL}$ family of DLs

We next define the problem of answering CQs over knowledge bases formulated in the $\mathcal{EL}$ family of DLs, which is the basic problem studied in this thesis.

Let $\mathcal{K} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$ be an $\mathcal{ELRO}^+_\bot$ KB and let $q = \exists \vec{y}.\, \psi(\vec{x}, \vec{y})$ be a CQ. Then q is *over* $\mathcal{K}$ if q uses only the atomic concepts in $\mathsf{con}_{\mathcal{K}}$, the roles in $\mathsf{rol}_{\mathcal{K}}$, and the individuals in $\mathsf{ind}_{\mathcal{K}}$. A substitution π is a *certain answer* to q over $\mathcal{K}$, written $\mathcal{K} \models_{\mathcal{DL}} \pi(q)$, if $\mathsf{dom}(\pi) = \vec{x}$, each element in $\mathsf{rng}(\pi)$ is an individual in $\mathsf{ind}_{\mathcal{K}}$, and $\mathcal{I} \models \pi(q)$ for each DL model $\mathcal{I}$ of $\mathcal{K}$. When q is a Boolean CQ, π being a certain answer to q over $\mathcal{K}$ implies that $\pi = \emptyset$ and $\pi(q) = q$; so we simply write $\mathcal{K} \models_{\mathcal{DL}} q$.

Instead of using the standard logical consequence symbol $\models$, we use the symbol $\models_{\mathcal{DL}}$ to stress that the certain answers to q over $\mathcal{K}$ are defined in terms of description logic models. In contrast, the certain answers to q over a rule base Σ are defined in terms of first-order models, and so we write $\Sigma \models \pi(q)$.

Answering q over $\mathcal{K}$ amounts to computing the set of all certain answers to q over $\mathcal{K}$. As stated, CQ answering is a function problem; in this thesis, however, we study the complexity of the associated decision problem, called the *recognition problem*, which decides whether $\mathcal{K} \models_{\mathcal{DL}} \pi(q)$. When q is an instance query, the recognition problem is also known as *instance checking*. Following Vardi [100], both q and $\mathcal{K}$ are considered as part of the input for the *combined complexity*, only $\mathcal{K}$ is considered as part of the input for the *knowledge base complexity*, and only the ABox $\mathcal{A}$ is considered as part of the input for the *data complexity*.

3.2.1 Decidability of CQ answering over $\mathcal{ELRO}^+_\bot$ via Regularity

When the RBox component is not restricted to a role hierarchy, Krisnadhi and Lutz [63], Krötzsch et al. [66], and Rosati [86] independently showed that CQ answering over $\mathcal{ELRO}^+_\bot$ KBs is undecidable. Intuitively, unrestricted role inclusions can 'simulate' derivations in context-free grammars; thus, a CQ can check whether two context-free languages have a non-empty intersection, which is known to be undecidable [50]. To regain decidability, we next recapitulate the definition of so-called regular RBoxes by Horrocks and Sattler [51].

Let $\mathcal{R}$ be an $\mathcal{ELRO}^+_\bot$ RBox and let $\prec$ be the smallest transitive relation on $\mathsf{rol}_\mathcal{R}$ such that, for each role inclusion $R_1 \cdots R_n \sqsubseteq R \in \mathcal{R}$ with $R \neq \top_r$ and each $i \in [1..n]$ with $R_i \neq R$, we have that $R_i \prec R$. RBox $\mathcal{R}$ is *regular* if $\prec$ is irreflexive and each role inclusion $\rho \sqsubseteq R$ in $\mathcal{R}$ with $R \neq \top_r$ is of the form

(t1) $\epsilon \sqsubseteq R$,

(t2) $R \cdot R \sqsubseteq R$,

(t3) $R_1 \cdots R_n \cdot R \sqsubseteq R$ where $n \geq 0$ and $R_i \neq R$ for each $i \in [1..n]$,

(t4) $R_1 \cdots R_n \sqsubseteq R$ where $n \geq 1$ and $R_i \neq R$ for each $i \in [1..n]$, or

(t5) $R \cdot R_1 \cdots R_n \sqsubseteq R$ where $n \geq 0$ and $R_i \neq R$ for each $i \in [1..n]$.

By induction on $\prec$, we next define the *level* $\mathsf{lv}(R)$ of each role $R \in \mathsf{rol}_{\mathcal{R}}$ as follows: $\mathsf{lv}(R) = 0$ if no role $S \in \mathsf{rol}_{\mathcal{R}}$ exists such that $S \prec R$; otherwise, $\mathsf{lv}(R) = 1 + \max\{\mathsf{lv}(S) \mid S \prec R\}$. Clearly, $\mathsf{lv}(R)$ can be computed in time polynomial in $|\mathcal{R}|$.

Horrocks and Sattler [51] showed that, for each regular RBox $\mathcal{R}$ and each role $R \in \mathsf{rol}_{\mathcal{R}}$, the language $\mathcal{L}(R)$ is regular. Hence, the undecidability results by Krisnadhi and Lutz [63], Krötzsch et al. [66], and Rosati [86] do not apply to CQ answering over $\mathcal{ELRO}^{+}_{\bot}$ knowledge bases with regular RBoxes. In particular, Ortiz et al. [83] showed that the problem is in 2ExpTime in combined complexity. Role hierarchies are generally not regular RBoxes, so this result does not immediately transfer to $\mathcal{ELHO}^{+}_{\bot}$. Nevertheless, Ortiz et al. [83] also showed that CQ answering over $\mathcal{ELHO}^{+}_{\bot}$ knowledge bases is decidable and in ExpTime in combined complexity.

The EL profile of OWL 2 imposes a syntactic restriction on role inclusions that extends the regularity restriction by Horrocks and Sattler [51]. Unfortunately, in Chapter 12 we show that the regularity condition in the OWL 2 specification is flawed: it does not ensure the regularity of the languages induced by role inclusions. For this reason, in the rest of this thesis, we consider only knowledge bases in which the RBox component is regular. In Chapter 12, we then propose a revised version of the syntactic restriction for OWL 2 and present so-called weakly regular RBoxes which generalise regular RBoxes and extend to role hierarchies. We also show that each knowledge base with a weakly regular RBox can be transformed in polynomial time into a knowledge base with a regular RBox without affecting the answers to CQs; thus, all our results carry over to knowledge bases in which the RBox is weakly regular, including role hierarchies.

3.2.2 Normalising Knowledge Bases

All the CQ answering algorithms that we present in this thesis require that each knowledge base $\mathcal{K} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$ is *normalised*, which is the case if the following conditions hold.

Table 3.2: Rewrite rules for $\mathcal{ELRO}^+_\bot$ TBoxes and RBoxes

	$\mathsf{key}(C, R_1, \ldots, R_n)$	$\mapsto$	$\{C \sqsubseteq X_C,$	$\mathsf{key}(X_C, R_1, \ldots, R_n)\}$
	$\mathsf{range}(R, C)$	$\mapsto$	$\{\mathsf{range}(R, X_C),$	$X_C \sqsubseteq C\}$
	$C \sqsubseteq D$	$\mapsto$	$\{C \sqsubseteq X_C,$	$X_C \sqsubseteq D\}$
	$E \sqcap \{a\} \sqsubseteq A$	$\mapsto$	$\{X_{\{a\}}(a),$	$E \sqcap X_{\{a\}} \sqsubseteq A\}$
	$C \sqcap A_1 \sqsubseteq A$	$\mapsto$	$\{C \sqsubseteq X_C,$	$X_C \sqcap A_1 \sqsubseteq A\}$
	$A \sqsubseteq C \sqcap D$	$\mapsto$	$\{A \sqsubseteq C,$	$A \sqsubseteq D\}$
	$\exists R.C \sqsubseteq A$	$\mapsto$	$\{C \sqsubseteq X_C,$	$\exists R.X_C \sqsubseteq A\}$
	$A \sqsubseteq \exists R.C$	$\mapsto$	$\{A \sqsubseteq \exists R.X_C,$	$X_C \sqsubseteq C\}$
($t3$)	$R_1 \cdots R_n \cdot R \sqsubseteq R$	$\mapsto$	$\{S \cdot R \sqsubseteq R,$	$R_1 \cdots R_n \sqsubseteq S\}$
($t4$)	$R_1 \cdots R_n \sqsubseteq R$	$\mapsto$	$\{S \cdot R_n \sqsubseteq R,$	$R_1 \cdots R_{n-1} \sqsubseteq S\}$
($t5$)	$R \cdot R_1 \cdots R_n \sqsubseteq R$	$\mapsto$	$\{R \cdot S \sqsubseteq R,$	$R_1 \cdots R_n \sqsubseteq S\}$

(n1) Each axiom in $\mathcal{T}$ is in one of the following forms, for $A_{(i)} \in N_\mathfrak{C}$, $a \in N_\mathfrak{I}$, and $R \in N_\mathfrak{R}$.

$$A_1 \sqsubseteq A \qquad A_1 \sqcap A_2 \sqsubseteq A \qquad \exists R.A_1 \sqsubseteq A \qquad \exists R.\mathsf{Self} \sqsubseteq A$$

$$A_1 \sqsubseteq \{a\} \qquad A_1 \sqsubseteq \exists R.A \qquad A_1 \sqsubseteq \exists R.\mathsf{Self} \qquad \mathsf{key}(A, R_1, \ldots, R_n)$$

(n2) Each role in $\mathcal{T} \cup \mathcal{A}$ also occurs in $\mathcal{R}$, each role inclusion $\rho \sqsubseteq R \in \mathcal{R}$ is such that $|\rho| \leq 2$ and $R \neq \top_r$, and $\mathcal{R}$ does not contain role inclusions of the form $R \sqsubseteq R$.

(n3) ABox $\mathcal{A}$ is non-empty.

Note that normalised knowledge bases do not contain range restriction. We next show that each $\mathcal{ELRO}^+_\bot$ knowledge base can be normalised in linear time without affecting the regularity of the RBox component or the answers to CQs.

Proposition 3.1. *For each $\mathcal{ELRO}^+_\bot$ KB $\mathcal{K} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$ where $\mathcal{R}$ is a regular RBox, one can compute in linear time a normalised KB $\mathcal{K}' = \mathcal{T}' \cup \mathcal{R}' \cup \mathcal{A}'$ such that*

- *$\mathcal{R}'$ is regular, and*
- *for each CQ q over $\mathcal{K}$ and each substitution π, $\mathcal{K} \models_{\mathcal{DL}} \pi(q)$ if and only if $\mathcal{K}' \models_{\mathcal{DL}} \pi(q)$.*

Proof. Let $\mathcal{K} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$ be an $\mathcal{ELRO}^+_\bot$ KB $\mathcal{K}$ where $\mathcal{R}$ is a regular RBox, let q be a CQ over $\mathcal{K}$, and let π be a substitution.

We satisfy property (n1) in two steps. First, we construct a KB $\mathcal{K}_1$ such that each TBox axiom in $\mathcal{K}_1$ is either in normal form or a range restriction of the form $\mathsf{range}(R, A)$ with $R \in N_{\mathfrak{R}}$ and $A \in N_{\mathfrak{C}}$. To this end, for each concept C occurring in $\mathcal{K}$, let X_C be a fresh atomic concept uniquely associated with C. Then, KB $\mathcal{K}_1$ is obtained from $\mathcal{K}$ by exhaustively decomposing each TBox axiom in $\mathcal{K}$ that is not already in normal form using the rewrite rules shown in the upper part Table 3.2, where $A_{(i)} \in N_{\mathfrak{C}}$, $a \in N_{\mathfrak{I}}$, E is a possibly empty conjunction of concepts, C and D are concepts, and $R_{(i)} \in N_{\mathfrak{R}}$. It is well-known [4, 64] that only linearly many rewrite steps are required to compute $\mathcal{K}_1$ and that $\mathcal{K} \models_{\mathcal{DL}} \pi(q)$ if and only if $\mathcal{K}_1 \models_{\mathcal{DL}} \pi(q)$.

Next, for each role $R \in \mathsf{rol}_{\mathcal{K}_1}$ and each atomic concept $A \in \mathsf{con}_{\mathcal{K}_1}$, let $X_{R,A}$ be a fresh atomic concept uniquely associated with R and A. Then $\mathcal{K}_2$ is obtained from $\mathcal{K}_1$ by carrying out the following operations for each axiom $\mathsf{range}(S, B) \in \mathcal{K}_1$ and each role R with $R \sqsubseteq^*_{\mathcal{R}} S$:

- replace each axiom of the form $A_1 \sqsubseteq \exists R.A \in \mathcal{K}_1$ with three axioms $A_1 \sqsubseteq \exists R.X_{R,A}$, $X_{R,A} \sqsubseteq A$, and $X_{R,A} \sqsubseteq B$ in $\mathcal{K}_2$;
- for each axiom $A_1 \sqsubseteq \exists R.\mathsf{Self} \in \mathcal{K}_1$, add an axiom $A_1 \sqsubseteq B \in \mathcal{K}_2$;
- if $\epsilon \sqsubseteq R \in \mathcal{K}_1$ or $R = \top_r$, add an axiom $\top_c \sqsubseteq B \in \mathcal{K}_2$;
- remove the range restriction $\mathsf{range}(S, B)$ from $\mathcal{K}_2$.

The resulting KB $\mathcal{K}_2$ satisfies (n1) and can be computed in time linear in $|\mathcal{K}_1|$. Baader et al. [5, Lemma 1] showed that $\mathcal{K}_1 \models_{\mathcal{DL}} \pi(q)$ if and only if $\mathcal{K}_2 \models_{\mathcal{DL}} \pi(q)$.

We next satisfy property (n2). Let $\mathcal{K}_3$ be the result of exhaustively decomposing each role inclusion $\rho \sqsubseteq R$ with $|\rho| > 2$ occurring in $\mathcal{K}_2$ using the rewrite rules (t3)–(t5) in the lower part of Table 3.2, where each $R_i \neq R$ and each occurrence of role S is fresh. Only linearly many rewrite steps are required to satisfy (n2), and the resulting RBox is regular. Furthermore, each model of $\mathcal{K}_3$ is also a model of $\mathcal{K}_2$ and each model $\mathcal{I}$ of $\mathcal{K}_2$ can be expanded to a model $\mathcal{J}$ of $\mathcal{K}_3$ by interpreting each fresh role S occurring in $\mathcal{K}_3 \setminus \mathcal{K}_2$ as $(S)^{\mathcal{J}} = (\rho_S)^{\mathcal{J}}$, where ρ_S is the unique role chain such that $\rho_S \sqsubseteq S$ occurs in $\mathcal{K}_3$. Thus, we have $\mathcal{K}_2 \models_{\mathcal{DL}} \pi(q)$ if and only if $\mathcal{K}_3 \models_{\mathcal{DL}} \pi(q)$.

Next, let $\mathcal{K}_4$ be the result of removing all axioms of the forms $\rho \sqsubseteq \top_r$ and $R \sqsubseteq R$ in $\mathcal{K}_3$; all removed axioms are tautologies, so we have $\mathcal{K}_3 \models_{\mathcal{DL}} \pi(q)$ if and only if $\mathcal{K}_4 \models_{\mathcal{DL}} \pi(q)$.

KB $\mathcal{K}_5$ is the result of adding an axiom $\bot_r \sqsubseteq R$, for each role R that occurs in $\mathcal{K}_4$ but does not occur in its RBox component. The axioms in $\mathcal{K}_5 \setminus \mathcal{K}_4$ preserve regularity and are tautologies, so $\mathcal{K}_4 \models_{\mathcal{DL}} \pi(q)$ if and only if $\mathcal{K}_5 \models_{\mathcal{DL}} \pi(q)$.

Finally, we satisfy property (n3). For a' a fresh individual not occurring in $\mathcal{K}_5$, let $\mathcal{K}'$ be the result of adding $\top_c(a')$ to $\mathcal{K}_5$. Assertion $\top_c(a')$ is a tautology, so $\mathcal{K}_5 \models_{\mathcal{DL}} \pi(q)$ if and only if $\mathcal{K}' \models_{\mathcal{DL}} \pi(q)$, as required. □

Chapter 4

The CQ Answering Algorithms at a Glance

In this thesis, we present novel complexity results, as well as novel algorithms for answering expressive queries over knowledge bases expressed in important members of the $\mathcal{EL}$ family of DLs. In particular, in Chapters 6 and 9, we introduce two novel worst-case optimal algorithms for answering conjunctive queries over $\mathcal{ELHO}_\bot^+$ and $\mathcal{ELRO}_\bot^+$ knowledge bases, respectively. The two algorithms are conceptually similar, so in this chapter we present the intuitions underlying the two procedures.

Each $\mathcal{ELRO}_\bot^+$ knowledge base $\mathcal{K}$ can be translated into an equivalent rule base. Hence, a conjunctive query q over $\mathcal{K}$ can be answered by evaluating q over a so-called *universal interpretation*—a possibly infinite set of assertions that, when seen as an interpretation, can be homomorphically embedded into any other model of $\mathcal{K}$. Universal interpretations are usually obtained by applying one of the many adaptations of the chase procedure [2, 74, 75, 12]. In this thesis, we first translate $\mathcal{K}$ into a rule base $\Xi_\mathcal{K}$ and then obtain universal interpretations of $\mathcal{K}$ by using our variant of the oblivious chase.

Each universal interpretation I of $\Xi_\mathcal{K}$ can be viewed as a directed graph that contains an edge $\langle w, w' \rangle$ labelled by role R for each binary assertion $R(w, w')$ occurring in I. Moreover, each labelled null occurring in I can be uniquely associated with an axiom of the form $A_1 \sqsubseteq \exists R.A$ that was used to generate it, and we call the pair R, A the element's *type*.

Our translation of $\mathcal{K}$ into a rule base $\Xi_{\mathcal{K}}$ is slightly unusual because the rules in $\Xi_{\mathcal{K}}$ not only capture the axioms in $\mathcal{K}$, but also capture additional consequences that describe the structural properties of $\mathcal{K}$. We use these consequences to distinguish between three types of edges in the directed graph associated with I.

The first type of edges, called *direct edges*, are those edges of I that either point to named individuals from $N_{\mathfrak{I}}$, or are obtained using axioms of the form $A_1 \sqsubseteq \exists R.A$ from $\mathcal{K}$. In our transformation, we identify direct edges obtained using an axiom of the form $A_1 \sqsubseteq \exists R.A$ by associating each role R with the *direct predicate* $\mathbb{D}_R$ for R, and then by transforming $A_1 \sqsubseteq \exists R.A$ into the existential rule $A_1(x) \rightarrow \exists z.R(x,z) \wedge \mathbb{D}_R(x,z) \wedge A(z)$. As direct edges can be propagated by simple role inclusions, we also add a rule $\mathbb{D}_S(x,y) \rightarrow \mathbb{D}_R(x,y)$ for each axiom $S \sqsubseteq R$ in $\mathcal{K}$. Then, an edge $\langle w, w' \rangle$ labelled by role R is a *direct edge* in I if w' is a named individual or $\mathbb{D}_R(w,w')$ occurs in I. When one restricts I to only those edges that are direct for some role R, interpretation I resembles a family of directed trees whose roots are the individuals in $\mathcal{K}$ and whose (direct) edges point from parents to children or to the individuals in $\mathcal{K}$.

The second type of edges, called *self edges*, are the loops in I that either connect named individuals to themselves, or are obtained using self-restrictions or reflexive roles in $\mathcal{K}$. We identify the edges obtained using self-restrictions or reflexive roles by associating each role R with a self predicate $\mathbb{S}_R$ for R, and then by transforming each axiom of the form $A \sqsubseteq \exists R.\mathsf{Self}$ into the rule $A_1(x) \rightarrow R(x,x) \wedge \mathbb{S}_R(x)$ and each axiom $\epsilon \sqsubseteq R$ into the rule $\top_c(x) \rightarrow R(x,x) \wedge \mathbb{S}_R(x)$. We also add a rule $\mathbb{S}_S(x) \rightarrow \mathbb{S}_R(x)$ for each axiom $S \sqsubseteq R$ in $\mathcal{K}$; hence, the definition of self edges propagates through simple role inclusions. Then, a loop $\langle w, w \rangle$ labelled by role R is a *self edge* in I if w is a named individual or $\mathbb{S}_R(w)$ occurs in I.

The third type of edges, called *composite edges*, are those labelled edges in I that are neither direct nor self edges. Then, an edge $\langle w, w' \rangle$ labelled by role R is a *composite edge* in I if w' is a labelled null, $\mathbb{D}_R(w,w')$ does not occur in I, and $w = w'$ implies that $\mathbb{S}_R(w)$ does not occur in I. It follows that each composite edge must be obtained using a role inclusion $R_1 \cdots R_n \sqsubseteq R$ with $n > 1$ occurring in $\mathcal{K}$.

Examples 4.1 and 4.2 illustrate the direct, self, and composite edges of a universal interpretation of $\Xi_{\mathcal{K}}$.

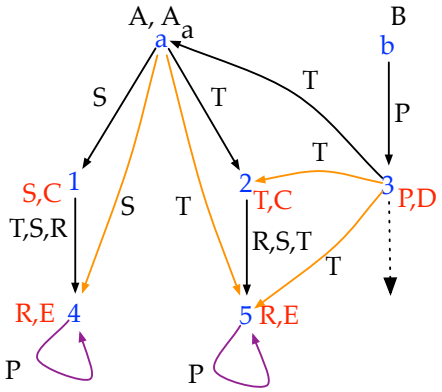

Figure 4.1: The universal interpretation for the KB in Example 4.1

Example 4.1. *Consider the knowledge base $\mathcal{K}$ whose ABox $\mathcal{A}$ contains two unary assertions $A(a)$ and $B(b)$, and whose TBox $\mathcal{T}$ consists of the following axioms.*

$$A \sqsubseteq \exists S.C \qquad B \sqsubseteq \exists P.D$$

$$A \sqsubseteq \exists T.C \qquad D \sqsubseteq \exists P.D$$

$$C \sqsubseteq \exists R.E \qquad D \sqsubseteq \exists T.A_a$$

$$E \sqsubseteq \exists P.\mathsf{Self} \qquad A_a \sqsubseteq \{a\}$$

Furthermore, the RBox $\mathcal{R}$ of $\mathcal{K}$ contains the following role inclusions.

$$R \sqsubseteq S \tag{4.1}$$

$$S \cdot S \sqsubseteq S \tag{4.2}$$

$$R \sqsubseteq T \tag{4.3}$$

$$T \cdot T \sqsubseteq T \tag{4.4}$$

Figure 4.1 shows a universal interpretation I of $\mathcal{K}$. Atoms involving $\top_c$ and $\top_r$ are not shown for clarity. Each labelled null is represented using a number; the element's type is shown in red. Direct edges are black, self edges are purple, and composite edges are orange. Axiom $D \sqsubseteq \exists P.D$ in $\mathcal{T}$ makes I infinite; in the figure, we encode the infinitely many successor of 3 of type P, D using a black dotted edge. Each such successor has a T-labelled direct edge to a, and two T-labelled composite edges to 2 and 5, respectively. Since

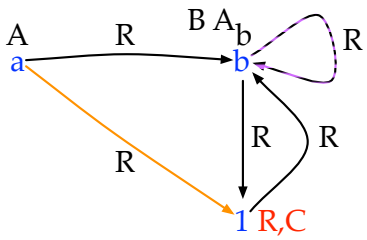

Figure 4.2: The universal interpretation for the KB in Example 4.2

I is infinite, a terminating algorithm cannot materialise I and then evaluate CQs in it.

Example 4.2. *Consider the knowledge base $\mathcal{K}$ whose ABox $\mathcal{A}$ contains $A(a)$, $B(b)$, and $R(a,b)$, whose RBox contains $R \cdot R \sqsubseteq R$, and whose TBox $\mathcal{T}$ contains the following axioms.*

$$B \sqsubseteq \exists R.C \qquad\qquad C \sqsubseteq \exists R.A_b \qquad\qquad A_b \sqsubseteq \{b\}$$

Figure 4.2 shows a universal interpretation I of $\mathcal{K}$. The notation is as explained in Example 4.1; edges that are both self and direct are shown using a black and purple solid line.

Edges $R(b,1)$ and $R(1,b)$ are obtained by applying the axioms in the TBox $\mathcal{T}$, and so their are direct and labelled with the direct predicate $\mathbb{D}_R$. Edges $R(a,b)$ and $R(b,b)$ point towards named individuals from $N_\mathfrak{I}$, and so their are also direct. Moreover, edge $R(b,b)$ loops on a named individual, so it is also a self edge. Finally, $R(a,1)$ is the only edge that points to a labelled null and is neither self nor direct; that is, $R(a,1)$ is the only composite edge. Thus, the direct and self predicates allow us to partition the edges in I that point to labelled nulls *into direct, self, and composite edges. In contrast, the* edges in I that point to named individuals *can either be direct or self edges, or both.*

In Section 4.1, we review the existing approaches to answering CQs over DL knowledge bases and discuss why these techniques cannot be easily adapted to obtain worst-case optimal and practical procedures for answering CQs over $\mathcal{EL}$ knowledge bases that contain nominals and role inclusions. Then, in Section 4.2 we discuss the intuitions behind our approach to answering conjunctive queries which we will formalise in Chapters 5, 6, and 9.

4.1 Existing Approaches to Answering CQs

Techniques for answering CQs over DL knowledge bases developed thus far can be broadly classified into the following four groups.

The first group consists of automata-based approaches for DLs such as Horn-$\mathcal{SHIQ}$ and Horn-$\mathcal{SROIQ}$ [83], $\mathcal{SH}$ [29], and the fragment of $\mathcal{ELRO}_\bot^+$ obtained by disallowing the universal role, reflexive roles, and self-restrictions [66]. All these approaches, however, require constructing automata whose size can be exponential in the knowledge base size, and are not practicable due to extensive don't-know nondeterminism.

The second group consists of rewriting-based approaches. Such techniques rewrite the query and/or the TBox into another formalism, usually a union of CQs or a datalog program; the relevant answers can then be obtained by evaluating the rewriting over the ABox seen as a first-order interpretation. Rewriting-based algorithms were initially proposed for members of the DL-Lite family [14, 20, 88, 55, 102], and later such algorithms have also been developed for the DLs $\mathcal{ELH}_\bot$ [86], $\mathcal{ELHIO}_\bot$ [85, 77] and Horn-$\mathcal{SHIQ}$ [30], members of the datalog$^\pm$ family [103], and existential rules [58], to name just a few. No rewriting approach, however, supports both nominals and (complex) role inclusions. Moreover, a common shortcoming is that rewritings can be exponential in the query and/or TBox size, so these approaches may also use exponential space. Although this is often not a problem in practice, such approaches are not worst-case optimal. An exception is the algorithm by Rosati [86] that rewrites an $\mathcal{ELH}_\bot$ TBox into a datalog program of polynomial size; however, the algorithm also uses a nondeterministic step to transform the CQ into a tree-shaped one, and it is not clear how to implement this step in a goal-directed manner.

The third group consists of hybrid approaches for answering CQs over knowledge bases expressed in Horn-$\mathcal{SROIQ}$ [107] and $\mathcal{SROIQ}$ [108, 97]—the DLs underpinning OWL 2 DL [53]. These approaches combine fully fledged OWL 2 DL reasoners with more efficient reasoners for the EL, QL, and RL profiles of OWL 2. Query evaluation is then delegated as much as possible to the reasoners for the profile at hand, while the heavy-weight OWL 2 DL reasoner is only used to ensure completeness by checking those answers that depend on constructs not in the relevant profile. No existing profile reasoner can answer conjunctive

queries over knowledge bases that contain all the constructs of $\mathcal{ELHO}_\bot^+$ and $\mathcal{ELRO}_\bot^+$; hence, these techniques require the use of computationally expensive OWL 2 DL reasoners, which can use space exponential in the size of the knowledge base and are usually not optimised for handling large ABoxes.

The fourth group consists of the so-called combined approaches. These techniques are based on a particular interpretation of $\mathcal{K}$ that we call the *compact interpretation*. The compact interpretation can be materialised in time polynomial in $|\mathcal{K}|$, and it can be used to test the consistency of, and answer instance queries over $\mathcal{K}$ [4, 67, 60]. Moreover, each CQ that maps onto the universal interpretation maps onto the compact interpretation as well; however, the converse does not necessarily hold as the compact interpretation finitely approximates the possibly infinite models of $\mathcal{K}$. As a remedy, *combined approaches* evaluate the CQ in the compact interpretation, but filter the results to eliminate unsound answers. Such approaches have been developed for members of the DL-Lite [59, 60, 73] and the $\mathcal{EL}$ [72] families of DLs, and the datalog$^\pm$ family [41] of rule-based languages. These approaches differ on how they implement the filtration step. Lutz et al. [72], Kontchakov et al. [60], and Gottlob et al. [41] rewrite in polynomial time the conjunctive query into a first-order query such that evaluating the rewritten query over the compact interpretation does not produce unsound answers. In contrast, Lutz et al. [73] first evaluate the conjunctive query over the compact interpretation to obtain *candidate answers*, and then purge unsound candidate answers using a *filtering procedure*.

Apart from the combined approach by Lutz et al. [72] that is applicable to the $\mathcal{ELH}_\bot$ fragment of $\mathcal{ELRO}_\bot^+$, all other techniques are applicable to DLs and rule-based formalisms that are orthogonal to the $\mathcal{EL}$ family of DLs; so in the rest of this section we focus on the former. Example 4.3 illustrates the compact interpretation for $\mathcal{K}$ and shows that evaluating conjunctive queries over this interpretation may produce unsound answers.

Example 4.3. *Consider the KB $\mathcal{K}$ whose RBox $\mathcal{R}$ is empty, whose ABox $\mathcal{A}$ contains three*

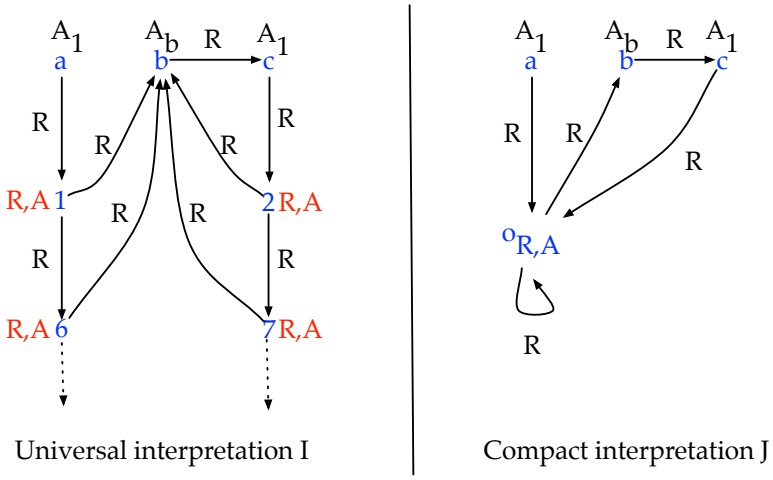

Figure 4.3: The universal and compact interpretations for the KB in Example 4.3

assertions $A_1(a)$, $A_1(c)$, *and* $R(b, c)$, *and whose TBox* $\mathcal{T}$ *contains the following axioms.*

$$A_1 \sqsubseteq \exists R.A \qquad A \sqsubseteq \exists R.A_b$$

$$A \sqsubseteq \exists R.A \qquad A_b \sqsubseteq \{b\}$$

Furthermore, let q_1, q_2, *and* q_3 *be the following BCQs.*

$$q_1 = \exists x.\ R(x, b) \tag{4.5}$$

$$q_2 = \exists x.\ R(a, x) \wedge R(c, x) \tag{4.6}$$

$$q_3 = \exists x, y.\ R(x, y) \wedge R(y, x) \tag{4.7}$$

The left part of Figure 4.3 shows a universal interpretation of $\mathcal{K}$*; the notation is as explained in Example 4.1. The compact interpretation for* $\mathcal{K}$ *is shown in the right part of Figure 4.3. It finitely approximates the universal interpretation by using so-called* auxiliary constants *of the form* $o_{R,A}$ *to represent all labelled nulls of type* R, A *occurring in* I*. Now query* q_1 *can be mapped onto both the compact and the universal interpretation, while queries* q_2 *and* q_3 *can be mapped only onto the compact interpretation. Thus, evaluating* q_2 *and* q_3 *over the compact interpretation produces* unsound answers.

To solve this problem, Lutz et al. [72] rewrite a CQ q over $\mathcal{K}$ into a first-order query q^* such that evaluating q^* over the compact interpretation generates the certain answers

to q over $\mathcal{K}$. The rewriting q^* is obtained by extending q with additional conditions that filter out unsound answers. Example 4.4 illustrates the different types of filtering conditions embedded in q^* using the queries from Example 4.3.

Example 4.4. *In their approach, Lutz et al. [72] incorporate two kinds of filtering conditions into the rewritten query which correspond to the two reasons why mapping CQs onto the compact interpretation may produce unsound answers.*

The first reason is best explained using CQ q_2 from Example 4.3. Atoms $R(a, x) \wedge R(c, x)$ in q_2 form a 'fork'—that is, two binary atoms that connect distinct terms with the same existential variable. Moreover, this 'fork' can only be satisfied in the compact interpretation by mapping the atoms to edges pointing towards auxiliary constant $o_{R,A}$. Constant $o_{R,A}$, however, represents the two labelled nulls 1 and 2 in I of type R, A that are connected via direct edges to a and c, respectively. The direct edges in I induce a forest-shaped interpretation; so, if we embed q_2 onto the universal interpretation by mapping x to, say, labelled null 1, then individuals a and c must be the unique predecessor of this element in I—that is, a and c must be equal in I. This condition is incorporated into the CQ by asserting that 'if x is mapped to an auxiliary constant, then a equals c'; so the resulting rewritten query q_2^* cannot be mapped onto the compact interpretation.

The second reason is best explained using query q_3 from Example 4.3. This query contains two atoms $R(x, y) \wedge R(y, x)$ that form a cycle. The compact interpretation contains a loop over auxiliary constant $o_{R,A}$ which allows us to embed q_3 into this interpretation by mapping both x and y to $o_{R,A}$. In contrast, the labelled nulls of type R, A that $o_{R,A}$ represents do not form loops in the universal interpretation. This condition is incorporated into the CQ by asserting that 'variables x and y cannot be mapped to auxiliary constants'; thus the resulting rewritten query q_3^* cannot be mapped onto the compact interpretation.

The knowledge base $\mathcal{K}$ from Example 4.3 contains nominals, a construct that is not supported in $\mathcal{ELH}_\bot$. Example 4.5 shows that nominals can interact with the filtering step by Lutz et al. [72]; so their approach is not directly applicable to $\mathcal{ELHO}_\bot^+$ and $\mathcal{ELRO}_\bot^+$ knowledge bases.

Example 4.5. *Let $\mathcal{K}$ be as in Example 4.3, and let q_4 and q_5 be as follows.*

$$q_4 = \exists x, y, z.\, R(a, y) \wedge R(c, z) \wedge R(y, x) \wedge R(z, x)$$

$$q_5 = \exists x, y, z.\, R(x, y) \wedge R(y, z) \wedge R(z, x)$$

Although queries q_4 and q_5 contain a 'fork' and a cycle, respectively, these queries can be mapped onto both the compact and the universal interpretations by mapping some of their atoms on the edges that connect labelled nulls with individuals in $\mathcal{K}$. These edges are obtained using the axiom $A_b \sqsubseteq \{b\}$, an axiom involving a nominal. Nominals are not in $\mathcal{ELH}_\perp$, so the filtering conditions on 'forks' and cycles by Lutz et al. [72] prohibit mapping queries q_4 and q_5 onto the compact interpretation and the results for these queries are thus incomplete.

To better explain how nominals interact with the filtering conditions on 'forks', we use query q_4. The rewriting procedure by Lutz et al. [72] extends this query with the following two constraints γ_1 and γ_2.

$$\gamma_1 = \textit{if } x \textit{ is mapped to auxiliary constant, then } y = z$$

$$\gamma_2 = \textit{if } y \textit{ is mapped to auxiliary constant, then } a = c$$

Constraint γ_1 is due to the fact that $R(y, x) \wedge R(z, x)$ is a 'fork', while constraint γ_2 is derived from a property specific to universal interpretations of $\mathcal{ELH}_\perp$ knowledge bases: that is, all edges in a universal interpretation of the KB are direct and either point from individuals to individuals, or from parents to children. Then, if y is mapped to an auxiliary constant, variable x is also mapped to an auxiliary constant, due to atom $R(y, x)$ in the query. But then, constraint γ_1 implies that $y = z$; so $R(a, y) \wedge R(c, z)$ is also a fork, and thus individuals a and c must be equal in I. The resulting first-order rewriting $q_4^* = q_4 \wedge \gamma_1 \wedge \gamma_2$ cannot be mapped onto the compact interpretation, and so the result for this query is incomplete. Roughly speaking, the reason for this is that nominals generate a new type of edge in the universal interpretation connecting labelled nulls with individuals, and these edges introduce dependencies between the filtering conditions: constraint γ_2 must to be applied only if

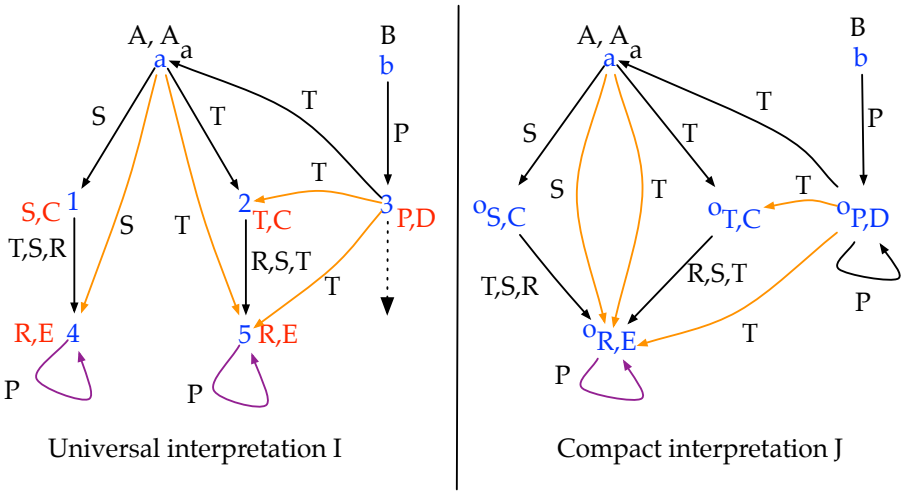

Figure 4.4: The universal and compact interpretations for the KB in Example 4.1

variable x is also mapped to an auxiliary constant.

To explain how nominals interact with the filtering conditions on cycles, we use query q_5. Variables x, y, and z participate in a cycle in q_5; so the rewriting procedure by Lutz et al. [72] incorporates the following constraint γ_3 into the query.

$$\gamma_3 = \textit{variables } x, y, \textit{ and } z \textit{ cannot be mapped to auxiliary constants}$$

The rewriting $q_5^* = q_5 \wedge \gamma_3$ cannot be mapped onto the compact interpretation, and so the result for this query is also incomplete. This is because the edges in the universal interpretation obtained using nominals can form cycles involving labelled nulls, and so the filtering condition in γ_3 is too restrictive.

In addition to nominals, Example 4.6 shows that transitive roles can also interact with the filtering conditions by Lutz et al. [72]; so their technique is unsuitable for knowledge bases that contain transitive roles, or more general forms of regular role inclusions.

Example 4.6. *Let $\mathcal{K}$ be as in Example 4.1; Figure 4.4 shows the universal and compact interpretations for $\mathcal{K}$. The universal interpretation is the same as the one shown in Figure 4.1, whereas the compact interpretation is obtained from the universal one by merging all labelled nulls of types R, E and P, D using auxiliary constants $o_{R,E}$ and $o_{P,D}$, respectively.*

Because $\mathcal{K}$ contains transitive roles, we next show that the technique by Lutz et al. [72]

can be unsound using the following conjunctive query q_6 over $\mathcal{K}$.

$$q_6 = \exists x.\, S(a, x) \wedge T(a, x)$$

Using Figure 4.4, one can see that query q_6 can be mapped solely onto the compact interpretation by mapping atoms $S(a, x)$ and $T(a, x)$ onto edges obtained using the transitivity of roles S and T. Moreover, query q_6 does not contain 'forks' and cycles, and so the rewriting step by Lutz et al. [72] does not incorporate filtering conditions into the query; so the rewritten query maps onto the compact interpretation thereby generating an unsound answer. This problem can be intuitively understood as follows. By 'unfolding' the query by (4.2) and (4.4), query q_6 essentially asks whether role chains $\rho_1 \in \mathcal{L}(S)$ and $\rho_2 \in \mathcal{L}(T)$ exist that label a path of direct edges in the universal interpretation I of $\mathcal{K}$ starting at a. In the compact interpretation, this is satisfied by $\rho_1 = S \cdot S$ and $\rho_2 = T \cdot T$ when x is mapped to constant $o_{R,E}$. Constant $o_{R,E}$, however, represents distinct labelled nulls 4 and 5 from I; hence, although 4 is connected to a via ρ_1 and 5 is connected to a via ρ_2, role chains ρ_1 and ρ_2 do not satisfy query q_6. In other words, the compact interpretation is 'too small' to represent the relevant conditions.

We next show that the technique by Lutz et al. [72] can also be incomplete using the following conjunctive query q_7 over $\mathcal{K}$.

$$q_7 = \exists x, y.\, R(x, y) \wedge S(a, y)$$

As opposed to query q_6, query q_7 can be mapped onto both the compact and the universal interpretation. Atoms $R(x, y) \wedge S(a, y)$, however, constitute a 'fork'; hence, the rewriting step by Lutz et al. [72] incorporates into the query a filtering condition asserting that 'if y is mapped to an auxiliary constant, then $x = a$'. The resulting rewritten query q_7^* cannot be mapped onto the compact interpretation; so the result for this query is incomplete.

4.2 Overview of our Approach

None of the existing combined query answering approaches support self-restrictions, nominals, transitive and reflexive roles, and more general forms of regular role inclusions. In our work, we bridge this gap and present two combined approaches for $\mathcal{ELHO}_\bot^+$ and $\mathcal{ELRO}_\bot^+$ KBs, respectively. Our techniques differ from the approach for $\mathcal{ELH}_\bot$ by Lutz et al. [72] in that we implement filtering using an external procedure à la Lutz et al. [73]: to answer a CQ, we first evaluate the query over the compact interpretation to obtain candidate answers and then filter out unsound candidate answers using a filtering procedure.

Following Krötzsch et al. [67], instead of directly materialising the compact interpretation, we capture this interpretation using a datalog program. In a similar way as with the translation of $\mathcal{K}$ into rules, we extend the known encoding by Krötzsch et al. [67] by capturing structural information about the knowledge base using direct and self predicates. Our encoding can be applied to any fragment of $\mathcal{ELRO}_\bot^+$ and the resulting datalog program can be directly used to test the consistency of the KB, as well as to answer instance queries.

Even though seemingly just a stylistic issue, a datalog specification of the compact interpretation may be beneficial in practice: one can either materialise all consequences of the program bottom-up in advance and obtain candidate answers by evaluating the CQ over the materialised compact interpretation, or one can use a top-down technique to compute candidate answers using only the part of the compact interpretation relevant for the query at hand. The latter can be particularly useful in informations systems that have read-only access to the data, or where data changes frequently.

After evaluating CQs over the datalog program, our algorithms use filtering procedures to filter out unsound candidate answers. Even though we present two separate filtering procedures for $\mathcal{ELHO}_\bot^+$ and $\mathcal{ELRO}_\bot^+$ knowledge bases, the checks these procedures use to determine the soundness of a candidate answer are conceptually the same.

Given a candidate answer, we check in polynomial time whether the answer satisfies an extended version of the filtering conditions on 'forks' and cycles by Lutz et al. [72]. Unlike the previously known conditions, our extended conditions account for nominals and regular role inclusions in the knowledge base. For candidate answers that do not map variables to

auxiliary constants, or are independent of the regular role inclusions in $\mathcal{K}$, these filtering conditions are sufficient to determine whether the candidate answer is sound.

For all other candidate answers, we nondeterministically construct a structure, called a *skeleton*, that finitely represents the infinitely many ways in which the candidate answer can be translated into a substitution mapping the query's variables to elements in the universal interpretation of $\mathcal{K}$.

The substitutions that the skeleton represents, however, are not guaranteed to satisfy those binary atoms in the CQ that the candidate answer maps onto edges obtained using role inclusions. So, we enrich the skeleton with additional constraints that express how these atoms need to be unfolded via role inclusions onto the universal interpretation. These constraints precisely capture the conditions for the soundness of the candidate answer: if the skeleton represents at least one substitution that satisfies the constraints, then the candidate answer is sound. Finally, we determine the existence of such a substitution using entailment checking over the direct and self predicates occurring in the datalog program.

Although the two filtering procedures implement similar checks, they differ in the way in which they formulate the constraints on the skeleton that express how binary atoms need to be unfolded via role inclusions. For $\mathcal{ELHO}_\bot^+$, we formulate these constraints directly on the roles occurring in the KB; for $\mathcal{ELRO}_\bot^+$, we instead encode the languages associated with each role in the KB using a bounded-stack PDA, and then we formulate the relevant constraints on these PDA. This difference is also reflected in the worst-case complexity of filtering: while the filtering procedure for $\mathcal{ELHO}_\bot^+$ runs in nondeterministic polynomial time, the procedure for $\mathcal{ELRO}_\bot^+$ can be implemented to use polynomial space. This is worst-case optimal: checking whether a candidate answer is sound is NP-complete and PSPACE-complete for $\mathcal{ELHO}_\bot^+$ and $\mathcal{ELRO}_\bot^+$, respectively. Moreover, to obtain goal-directed filtering, our procedures implement optimisations that reduce the number of nondeterministic choices to consider. Finally, both procedures exhibit pay-as-you-go behaviour: they run in polynomial time if the knowledge base is in $\mathcal{ELHO}_\bot$.

In Chapter 5, we present the datalog program that captures the compact interpretation of $\mathcal{K}$. In Chapters 6 and 9, we present the filtering procedures for $\mathcal{ELHO}_\bot^+$ and $\mathcal{ELRO}_\bot^+$, respectively; the latter procedure uses the PDA encoding of role inclusions from Chapter 8.

Chapter 5

Translating KBs into Datalog

The consequences of an $\mathcal{ELRO}_\bot^+$ knowledge base $\mathcal{K}$ that are relevant to determining the consistency of $\mathcal{K}$ and answering instance queries over $\mathcal{K}$ can be captured using a so-called compact interpretation of $\mathcal{K}$. Although this interpretation is a model of $\mathcal{K}$, it cannot be homomorphically embedded into every model of $\mathcal{K}$; hence, evaluating CQs on this interpretation can produce unsound answers. In this chapter, we present a datalog program $\mathsf{D}_\mathcal{K}$ that captures the compact interpretation of $\mathcal{K}$. We obtain $\mathsf{D}_\mathcal{K}$ by first translating $\mathcal{K}$ into a rule base $\Xi_\mathcal{K}$, whose universal interpretations can be used to evaluate conjunctive queries over $\mathcal{K}$, and then, following Krötzsch et al. [67], we translate $\Xi_\mathcal{K}$ into a datalog program $\mathsf{D}_\mathcal{K}$. Our translations differ from the existing ones in that we use the rules in $\Xi_\mathcal{K}$ and in $\mathsf{D}_\mathcal{K}$ to capture the structural properties of $\mathcal{K}$ using so-called direct and self predicates. We use the direct and self predicates in $\Xi_\mathcal{K}$ to emphasise that the universal interpretations of $\mathcal{K}$ resemble a forest-shaped structure, whereas in Chapters 6 and 9 we use the direct and self predicates in $\mathsf{D}_\mathcal{K}$ to develop filtering procedures that determine whether the answers obtained by evaluating CQs over $\mathsf{D}_\mathcal{K}$ are sound. For the rest of this chapter, we fix an $\mathcal{ELRO}_\bot^+$ KB $\mathcal{K} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$.

5.1 Translating Knowledge Bases into Rule Bases

We next show how to translate $\mathcal{K}$ into a rule base $\Xi_\mathcal{K}$. As well as being a first step towards transforming $\mathcal{K}$ into datalog, we use this translation to provide an alternative semantics

of $\mathcal{K}$ based on the universal interpretations of $\Xi_{\mathcal{K}}$—interpretations which can be obtained using our chase variant from Section 2.2. Hence, in the rest of this thesis, we identify the universal interpretations of $\mathcal{K}$ with those of $\Xi_{\mathcal{K}}$.

Our translation of $\mathcal{K}$ into rules is unusual in two respects.

First, we introduce an auxiliary unary predicate $\mathbb{C}$ and we uniquely associate each role $R \in \mathsf{rol}_{\mathcal{R}}$ with a direct predicate $\mathbb{D}_R$ and a self predicate $\mathbb{S}_R$. We use these predicates in our translation to capture the structural properties of $\mathcal{K}$; thus emphasising that each universal interpretation I of $\mathcal{K}$ resembles a family of directed trees as described in Chapter 4. In particular, the unary predicate $\mathbb{C}$ captures the individuals occurring in $\mathcal{K}$, the direct predicates allow us to identify those direct edges in I that point to labelled nulls, and the self predicates allow us to identify the self edges in I. Please note that, while the direct predicates label only the direct edges in I that point to labelled nulls, the self predicates label all self edges in I, including those pointing to named individuals; this allows us to translate axioms of the form $\exists R.\mathsf{Self} \sqsubseteq A$ into the first-order rule $\mathbb{S}_R(x) \rightarrow A(x)$.

Second, in our definition of first-order logic, we consider $\top_c$ and $\bot_c$, and $\top_r$ and $\bot_r$ as standard predicates, and so they are not assigned any predetermined meaning. In contrast, each DL interpretation interprets these predicates as shown in Table 3.1. We solve this semantic mismatch by directly axiomatising in $\Xi_{\mathcal{K}}$ the intended meaning of these predicates using additional datalog rules. This explicit axiomatisation of $\top_c$ and $\bot_c$, and $\top_r$ and $\bot_r$ into $\Xi_{\mathcal{K}}$ will provide practical benefits: the datalog translation $\mathsf{D}_{\mathcal{K}}$ of $\mathcal{K}$ inherits these additional rules from $\Xi_{\mathcal{K}}$, so we can evaluate $\mathsf{D}_{\mathcal{K}}$ using an arbitrary datalog engine without making any additional assumption on the engine's internal interpretation of these predicates.

Definition 5.1. *Let $\mathbb{C}$ be a fresh unary predicate and, for each $R \in \mathsf{rol}_{\mathcal{R}}$, let $\mathbb{S}_R$ and $\mathbb{D}_R$ be a fresh unary predicate and a fresh binary predicate uniquely associated with R. Set $\Xi_{\mathcal{T},\mathcal{R}}$ contains the translation of each axiom in $\mathcal{T} \cup \mathcal{R}$ into a rule as shown in Table 5.1, and $\mathsf{cl}_{\mathcal{K}}$ contains the atoms and the rules in (5.1)–(5.8). The* rule base for $\mathcal{K}$ *is $\Xi_{\mathcal{K}} = \Xi_{\mathcal{T},\mathcal{R}} \cup \mathsf{cl}_{\mathcal{K}} \cup \mathcal{A}$.*

$$\mathbb{C}(a) \qquad \forall a \in \mathsf{ind}_{\mathcal{K}} \tag{5.1}$$

$$\mathbb{C}(x) \wedge R(x,x) \rightarrow \mathbb{S}_R(x) \qquad \forall R \in \mathsf{rol}_{\mathcal{K}} \tag{5.2}$$

Table 5.1: Translating $\mathcal{ELRO}^{+}_{\bot}$ axioms into first-order rules

	Axiom		*Rule*
(1)	$A_1 \sqsubseteq A$	$\rightsquigarrow$	$A_1(x) \rightarrow A(x)$
(2)	$A_1 \sqcap A_2 \sqsubseteq A$	$\rightsquigarrow$	$A_1(x) \wedge A_2(x) \rightarrow A(x)$
(3)	$A_1 \sqsubseteq \{a\}$	$\rightsquigarrow$	$A_1(x) \rightarrow x \approx a$
(4)	$A_1 \sqsubseteq \exists R.A$	$\rightsquigarrow$	$A_1(x) \rightarrow \exists z.R(x,z) \wedge \mathbb{D}_R(x,z) \wedge A(z)$
(5)	$\exists R.A_1 \sqsubseteq A$	$\rightsquigarrow$	$R(x,y) \wedge A_1(y) \rightarrow A(x)$
(6)	$A \sqsubseteq \exists R.\mathsf{Self}$	$\rightsquigarrow$	$A(x) \rightarrow R(x,x) \wedge \mathbb{S}_R(x)$
(7)	$\exists R.\mathsf{Self} \sqsubseteq A$	$\rightsquigarrow$	$\mathbb{S}_R(x) \rightarrow A(x)$
(8)	$\mathsf{key}(A, R_1, \ldots, R_n)$	$\rightsquigarrow$	$A(x) \wedge A(y) \wedge \mathbb{C}(x) \wedge \mathbb{C}(y) \wedge \bigwedge_i \mathbb{C}(z_i) \wedge R_i(x,z_i) \wedge R_i(y,z_i) \rightarrow x \approx y$
(9)	$\epsilon \sqsubseteq R$	$\rightsquigarrow$	$\top_c(x) \rightarrow R(x,x) \wedge \mathbb{S}_R(x)$
(10)	$R_1 \cdots R_n \sqsubseteq R$	$\rightsquigarrow$	$R_1(x_0,x_1) \wedge \ldots \wedge R_n(x_{n-1},x_n) \rightarrow R(x_0,x_n)$

$$\mathbb{S}_S(x) \rightarrow \mathbb{S}_R(x) \qquad \forall S \sqsubseteq R \in \mathcal{R} \tag{5.3}$$

$$\mathbb{D}_S(x,y) \rightarrow \mathbb{D}_R(x,y) \qquad \forall S \sqsubseteq R \in \mathcal{R} \tag{5.4}$$

$$A(x) \rightarrow \top_c(x) \qquad \forall A \in \{\mathbb{C}\} \cup \mathsf{con}_{\mathcal{K}} \tag{5.5}$$

$$R(x,y) \rightarrow \top_c(x) \wedge \top_c(y) \qquad \forall R \in \mathsf{rol}_{\mathcal{K}} \tag{5.6}$$

$$\top_c(x) \wedge \top_c(y) \rightarrow \top_r(x,y) \tag{5.7}$$

$$\bot_r(x,y) \rightarrow \bot_c(x) \tag{5.8}$$

Example 5.1 shows that, due to axioms of type (4) from Table 5.1, rule base $\Xi_{\mathcal{K}}$ is not necessarily a datalog program.

Example 5.1. *Let $\mathcal{K}$ be the knowledge base from Example 4.1 on page 39. Then, the axioms in the TBox $\mathcal{T}$ of $\mathcal{K}$ are translated into rules as follows.*

$$A(x) \rightarrow \exists z.S(x,z) \wedge \mathbb{D}_S(x,z) \wedge C(z) \qquad E(x) \rightarrow P(x,x) \wedge \mathbb{S}_P(x)$$

$$A(x) \rightarrow \exists z.T(x,z) \wedge \mathbb{D}_T(x,z) \wedge C(z) \qquad A_a(x) \rightarrow x \approx a$$

$$C(x) \rightarrow \exists z.R(x,z) \wedge \mathbb{D}_R(x,z) \wedge E(z)$$

$$B(x) \rightarrow \exists z.P(x,z) \wedge \mathbb{D}_P(x,z) \wedge D(z)$$

$$D(x) \rightarrow \exists z.P(x,z) \wedge \mathbb{D}_P(x,z) \wedge D(z)$$

$$D(x) \rightarrow \exists z.T(x,z) \wedge \mathbb{D}_T(x,z) \wedge A_a(z)$$

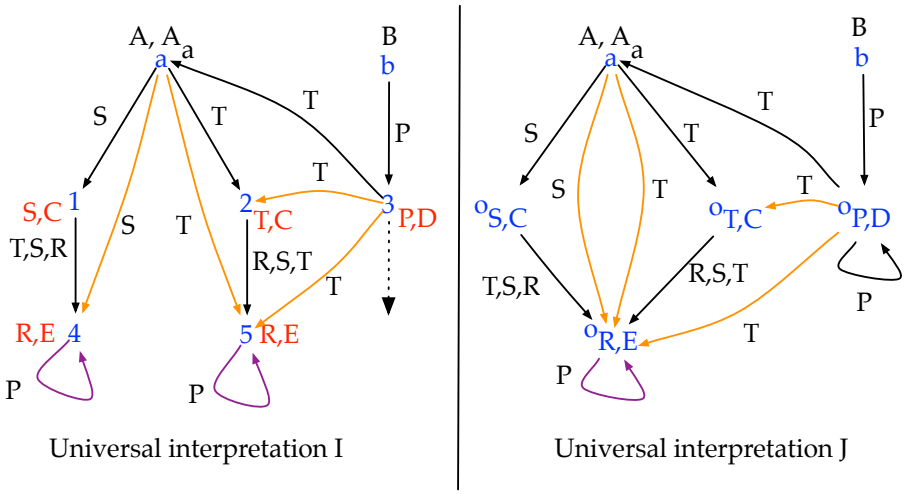

Figure 5.1: The universal interpretation I and J of $\Xi_{\mathcal{K}}$ and $\mathsf{D}_{\mathcal{K}}$ for the KB in Example 4.1

Rule base $\Xi_{\mathcal{K}}$ contains all the above rules and is thus not a datalog program. The left part of Figure 5.1 shows a universal interpretation I of $\Xi_{\mathcal{K}}$ obtained using our chase variant described in Section 2.2; atoms involving $\mathbb{C}$, $\top_c$, and $\top_r$ are not shown for clarity; notation is as explained in Example 4.1.

While our definition of $\Xi_{\mathcal{K}}$ differs from the usual translation of $\mathcal{K}$ into first-order logic [6], Proposition 5.2 shows that we can use $\Xi_{\mathcal{K}}$ to check the consistency of $\mathcal{K}$ and answer CQs over $\mathcal{K}$. When $\mathcal{K}$ is consistent, by Theorem 2.3 we can answer CQs over $\mathcal{K}$ by evaluating them in a universal interpretation of $\Xi_{\mathcal{K}}$. The proof of this result is given in Section 5.3.

Proposition 5.2. *$\Xi_{\mathcal{K}}$ satisfies the following properties for each CQ $q = \exists \vec{y}.\, \psi(\vec{x}, \vec{y})$ over $\mathcal{K}$ and each substitution π such that $\mathsf{dom}(\pi) = \vec{x}$ and each element in $\mathsf{rng}(\pi)$ occurs in $\mathsf{ind}_{\mathcal{K}}$.*

(1) $\mathcal{K}$ is inconsistent if and only if $\Xi_{\mathcal{K}} \models \exists y.\, \bot_c(y)$.

(2) $\mathcal{K} \models_{\mathcal{DL}} \pi(q)$ if and only if $\mathcal{K}$ is inconsistent or $\Xi_{\mathcal{K}} \models \pi(q)$.

We next formally characterise the structural properties of a universal interpretation I of $\mathcal{K}$ that we intuitively described in Chapter 4. To this end, please recall that each labelled null w occurring in I can be uniquely associated with an existential rule of the form $A_1(x) \rightarrow \exists z.\, R(x, z) \wedge \mathbb{D}_R(x, z) \wedge A(z)$ that was used to generate it, and that w's type

is R, A. Moreover, we let dir_I be the following relation on the terms occurring in inst_I.

$$\mathsf{dir}_I = \{\langle w, w' \rangle \mid \exists R \in \mathsf{rol}_{\mathcal{K}}, \exists A \in \mathsf{con}_{\mathcal{K}} : \mathbb{D}_R(w, w') \in \mathsf{inst}_I \wedge w' \in \Phi_N \text{ is of type } R, A\}$$

Hence, relation dir_I contains each edge $\langle w, w' \rangle$ of I such that $\langle w, w' \rangle$ is direct for some role R and w' is a labelled null.

In Lemma 5.3 we then show that relation dir_I is a forest rooted in the individuals occurring in $\Xi_{\mathcal{K}}$, which proves that the direct edges in I induce a forest-shaped structure in which all edges either point from parents to children or to the individuals in $\mathcal{K}$. Moreover, in property (2) we show that each atom $R(w, w')$ in I can be unfolded via the role inclusions in $\mathcal{K}$ into a path that consists only of direct and self edges. Finally, in properties (3) and (4) we focus on simple roles, the roles of $\mathcal{K}$ that cannot be expanded into role chains using role inclusions, and we show that each edge $\langle w, w' \rangle$ with $w' \in \Phi_N$ in I that is labelled by a simple role is either a self or a direct edge. The proof of this lemma is given in Section 5.3.

Lemma 5.3. *Interpretation I satisfies the following properties for all terms w and w' occurring in I and each role $R \in \mathsf{rol}_{\mathcal{K}}$.*

(1) Relation dir_I is a forest rooted in $N_{\mathfrak{I}}$.

(2) $R(w, w') \in \mathsf{inst}_I$ implies that a non-empty role chain $\rho = \chi_0 \cdot R_1 \cdot \chi_1 \cdots R_m \cdot \chi_m$ with $\rho \in \mathcal{L}(R)$ and terms $w_0, \ldots, w_m$ with $w_0 = w$ and $w_m = w'$ exist such that

(a) for each $i \in [1..m]$, either $w_i \in N_{\mathfrak{I}}$, or w_i is a labelled null of type R_i, A_i and $\mathbb{D}_{R_i}(w_{i-1}, w_i) \in \mathsf{inst}_I$.

(b) for each $i \in [1..m]$, we have that $R_i(w_{i-1}, w_i) \in \mathsf{inst}_I$; and

(c) for each $i \in [0..m]$ and each role S occurring in role chain χ_i, we have that $\{S(w_i, w_i), \mathbb{S}_S(w_i)\} \subseteq \mathsf{inst}_I$.

(3) $R(w, w) \in \mathsf{inst}_I$, R is simple, and $w \in \Phi_N$ imply that $\mathbb{S}_R(w) \in \mathsf{inst}_I$.

(4) $R(w, w') \in \mathsf{inst}_I$, R is simple, $w' \in \Phi_N$, and $w \neq w'$ imply that $\langle w, w' \rangle \in \mathsf{dir}_I$.

5.2 Translating Rule Bases into Datalog

To translate $\mathcal{K}$ into datalog, we approximate those rules in $\Xi_{\mathcal{K}}$ that contain existential quantifiers using the technique by Krötzsch et al. [67]: for each role $R \in \mathsf{rol}_{\mathcal{R}}$ and each atomic concept $A \in \mathsf{con}_{\mathcal{K}}$, we introduce a globally fresh and unique *auxiliary constant* $o_{R,A}$, which represents all the labelled nulls in the universal interpretation I of $\mathcal{K}$ of type R, A. In addition, unlike the existing datalog encoding by Krötzsch et al. [67], our datalog program $\mathsf{D}_{\mathcal{K}}$ inherits from $\Xi_{\mathcal{K}}$ the direct and self predicates which capture information about the structure of $\mathcal{K}$.

Definition 5.4. *Set N_{aux} contains a fresh constant $o_{R,A}$ not occurring in $N_{\mathfrak{I}}$ for each role R and each atomic concept A. Datalog program $\mathsf{D}_{\mathcal{T},\mathcal{R}}$ contains the translation of each axiom in $\mathcal{T} \cup \mathcal{R}$ of type other than (4) into a rule as shown in Table 5.1; furthermore, for each axiom $A_1 \sqsubseteq \exists R.A$ in $\mathcal{T}$, set $\mathsf{D}_{\mathcal{T},\mathcal{R}}$ contains $A_1(x) \rightarrow R(x, o_{R,A}) \wedge \mathbb{D}_R(x, o_{R,A}) \wedge A(o_{R,A})$. Then $\mathsf{D}_{\mathcal{K}} = \mathsf{D}_{\mathcal{T},\mathcal{R}} \cup \mathsf{cl}_{\mathcal{K}} \cup \mathcal{A}$ is the datalog program for $\mathcal{K}$.*

As for the universal interpretations of $\Xi_{\mathcal{K}}$, we view each universal interpretation J of $\mathsf{D}_{\mathcal{K}}$ as a directed graph that contains an edge $\langle u, u' \rangle$ labelled by role R for each binary assertion $R(u, u') \in J$. The direct and self predicates occurring in $\mathsf{D}_{\mathcal{K}}$ allow us to distinguish between direct, self, and composite edges in J. An edge $\langle u, u' \rangle$ labelled by role R in J is a *direct edge* if $u' \in N_{\mathfrak{I}}$ or $\mathbb{D}_R(u, u') \in J$, it is a *self edge* if $u = u'$ and $\mathbb{S}_R(u) \in J$, and it is a *composite edge* if u' is an auxiliary constant and the edge is neither direct nor self. Whenever two constants u and u' are connected by a direct predicate in J, we say that u is *directly connected* to u'.

Example 5.2 illustrates the different types of edges in a universal interpretation J of $\mathsf{D}_{\mathcal{K}}$.

Example 5.2. *The TBox axioms in the knowledge base $\mathcal{K}$ from Example 4.1 are translated*

into datalog as follows.

$$A(x) \to S(x, o_{S,C}) \wedge \mathbb{D}_S(x, o_{S,C}) \wedge C(o_{S,C}) \qquad E(x) \to S(x,x) \wedge \mathbb{S}_S(x)$$

$$A(x) \to T(x, o_{T,C}) \wedge \mathbb{D}_T(x, o_{T,C}) \wedge C(o_{T,C}) \qquad A_a(x) \to x \approx a$$

$$C(x) \to R(x, o_{R,E}) \wedge \mathbb{D}_R(x, o_{R,E}) \wedge E(o_{R,E})$$

$$B(x) \to P(x, o_{P,D}) \wedge \mathbb{D}_P(x, o_{P,D}) \wedge D(o_{P,D})$$

$$D(x) \to P(x, o_{P,D}) \wedge \mathbb{D}_P(x, o_{P,D}) \wedge D(o_{P,D})$$

$$D(x) \to T(x, o_{P,A_a}) \wedge \mathbb{D}_T(x, o_{P,A_a}) \wedge A_a(o_{P,A_a})$$

The right part of Figure 5.1 shows a universal interpretation J of $\mathsf{D}_\mathcal{K}$; atoms involving $\mathbb{C}$, $\top_c$, and $\top_r$ are not shown for clarity. We use the notation from Example 4.1 to distinguish between direct, self, and composite edges in J. Note that auxiliary constants o_{T,A_a} is 'merged' in J with individuals a, since $\mathsf{D}_\mathcal{K} \models o_{T,A_a} \approx a$.

By Definition 5.4, it is clear that we can compute the datalog program $\mathsf{D}_\mathcal{K}$ in time linear in the size of $\mathcal{K}$. Moreover, because we assume that $\mathcal{K}$ is normalised, each role inclusion $\rho \sqsubseteq R \in \mathcal{K}$ is such that $|\rho| \leq 2$. Hence, each rule in $\mathsf{D}_\mathcal{K}$ contains a fixed number of variables, and so we can compute the set of all consequences of $\mathsf{D}_\mathcal{K}$ in time polynomial in the size of $\mathcal{K}$ [24]. Proposition 5.5 summarises these results.

Proposition 5.5. *Program $\mathsf{D}_\mathcal{K}$ can be computed in time linear in $|\mathcal{K}|$; furthermore, the set of consequences of $\mathsf{D}_\mathcal{K}$ can be computed in time polynomial in $|\mathcal{K}|$.*

Example 5.2 shows that, due to equality rules, auxiliary constants in $\mathsf{D}_\mathcal{K}$ may be equal to individuals from $N_\mathfrak{I}$, thus not representing labelled nulls of I. Hence, in Definition 5.6, we introduce two sets: set $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ provides us with all auxiliary constants that are not equal to an individual from $N_\mathfrak{I}$, and set $\mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ provides us with a canonical representative for each individual from $N_\mathfrak{I}$ occurring in $\mathsf{D}_\mathcal{K}$.

Definition 5.6. *Let $<$ be a total order on constants such that $a < o_{R,A}$ for each individual $a \in N_\mathfrak{I}$ and each $o_{R,A} \in N_{aux}$. Set $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ contains each constant u occurring in $\mathsf{D}_\mathcal{K}$ for which no individual $a \in N_\mathfrak{I}$ exists such that $\mathsf{D}_\mathcal{K} \models u \approx a$. Furthermore, for each constant*

u occurring in $\mathsf{D}_\mathcal{K}$, let $u_\approx = u$ if $u \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$; otherwise, let $u_\approx$ be the $<$-smallest individual $a \in N_\mathfrak{I}$ such that $\mathsf{D}_\mathcal{K} \models u \approx a$. Set $\mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ contains $a_\approx$ for each individual $a \in \mathsf{ind}_\mathcal{K}$.

The next proposition shows that the datalog program $\mathsf{D}_\mathcal{K}$ can be used to test the consistency of, and answer instance queries over $\mathcal{K}$. By Proposition 5.5, both tasks can be done using polynomial time in the size of $\mathcal{K}$. The proof of Proposition 5.7 is given in Section 5.3.

Proposition 5.7. *Datalog program $\mathsf{D}_\mathcal{K}$ satisfies the following two properties for each atomic concept $A \in \mathsf{con}_\mathcal{K}$ and each individual $a \in \mathsf{ind}_\mathcal{K}$.*

- *$\mathcal{K}$ is inconsistent if and only if $\mathsf{D}_\mathcal{K} \models \exists y.\, \bot_c(y)$.*
- *$\mathcal{K} \models_{\mathcal{DL}} A(a)$ if and only if $\mathcal{K}$ is inconsistent or $\mathsf{D}_\mathcal{K} \models A(a)$.*

Datalog program $\mathsf{D}_\mathcal{K}$ can be seen as a strengthening of $\mathcal{K}$: all axioms of the form $A_1 \sqsubseteq \exists R.A$ in $\mathcal{K}$ are satisfied in a universal interpretation J of $\mathsf{D}_\mathcal{K}$ using a single auxiliary constant $o_{R,A}$. So, while we can use $\mathsf{D}_\mathcal{K}$ to compute the answers to instance queries over $\mathcal{K}$, evaluating a CQ q in J produces a set of candidate answers, ground substitutions that map all of q's variables to constants and embed q in J. For each certain answer π to q over $\mathcal{K}$, a candidate answer τ to q over $\mathsf{D}_\mathcal{K}$ exists such that $\pi \subseteq \tau$. The converse, however, does not necessarily hold; so in Definition 5.8 we call a candidate answer τ sound if it corresponds to a certain answer to q over $\mathcal{K}$.

Definition 5.8. *A substitution τ is a* candidate answer *to a CQ $q = \exists \vec{y}.\, \psi(\vec{x}, \vec{y})$ over $\mathsf{D}_\mathcal{K}$ if $\mathsf{dom}(\tau) = \vec{x} \cup \vec{y}$, each element of $\mathsf{rng}(\tau)$ is a constant occurring in $\mathsf{D}_\mathcal{K}$, and $\mathsf{D}_\mathcal{K} \models \tau(q)$. Such a candidate answer τ is* sound *if $\mathcal{K} \models_{\mathcal{DL}} \tau|_{\vec{x}}(q)$.*

In Chapters 6 and 9, we present two filtering procedures for $\mathcal{ELHO}_\bot^+$ and $\mathcal{ELRO}_\bot^+$, respectively, that check whether a candidate answer τ to a CQ q over $\mathsf{D}_\mathcal{K}$ is sound.

5.3 Proof of Correctness

In this section, we prove the technical results presented in this chapter. For convenience, in the rest of this thesis, we shall assume w.l.o.g. that each instance of the chase procedure presented in Section 2.2 is parametrised with the total order $<$ specified in Definition 5.6,

and that $a < w$ for each individual $a \in N_{\mathfrak{I}}$ and each labelled null $w \in \Phi_N$. Also, we let I and J be arbitrary universal interpretations of $\Xi_{\mathcal{K}}$ and $\mathsf{D}_{\mathcal{K}}$, respectively. Next, in Section 5.3.1 we prove the correctness of our encoding of $\mathcal{K}$ into rule bases, in Section 5.3.2 we formally characterise the structural properties of the universal interpretations of $\mathcal{K}$, and in Section 5.3.3 we show the correctness of our encoding of $\mathcal{K}$ into datalog.

5.3.1 Correctness of the Translation into Rule Bases

We show that our translation of $\mathcal{K}$ into a rule base $\Xi_{\mathcal{K}}$ preserves both knowledge base consistency and the answers to conjunctive queries.

Proposition 5.2. *$\Xi_{\mathcal{K}}$ satisfies the following properties for each CQ $q = \exists \vec{y}.\ \psi(\vec{x}, \vec{y})$ over $\mathcal{K}$ and each substitution π such that $\mathsf{dom}(\pi) = \vec{x}$ and each element in $\mathsf{rng}(\pi)$ occurs in $\mathsf{ind}_{\mathcal{K}}$.*

(1) $\mathcal{K}$ is inconsistent if and only if $\Xi_{\mathcal{K}} \models \exists y.\ \bot_c(y)$.

(2) $\mathcal{K} \models_{\mathcal{DL}} \pi(q)$ if and only if $\mathcal{K}$ is inconsistent or $\Xi_{\mathcal{K}} \models \pi(q)$.

Proof. To prove the proposition, we show that, for each CQ $q = \exists \vec{y}.\ \psi(\vec{x}, \vec{y})$ over $\mathcal{K}$ and each substitution π where $\mathsf{dom}(\pi) = \vec{x}$ and each element in $\mathsf{rng}(\pi)$ occurs in $\mathsf{ind}_{\mathcal{K}}$, we have that $\mathcal{K} \not\models_{\mathcal{DL}} \pi(q)$ if and only if $\Xi_{\mathcal{K}} \not\models \exists y.\ \bot_c(y)$ and $\Xi_{\mathcal{K}} \not\models \pi(q)$. Since $\mathcal{K}$ is consistent if and only if $\mathcal{K} \not\models_{\mathcal{DL}} \exists y.\ \bot_c(y)$, this shows that properties (1) and (2) hold. For the rest of this proof, let q and π be an arbitrary CQ and an arbitrary substitution as specified above.

($\Rightarrow$) Assume that $\mathcal{K} \not\models_{\mathcal{DL}} \pi(q)$; hence, a description logic model $\mathcal{I}$ of $\mathcal{K}$ exists such that $\mathcal{I} \not\models \pi(q)$. Please recall that every description logic interpretation is also a first-order interpretation. We next expand $\mathcal{I}$ to a first-order interpretation $\mathcal{J}$ of $\Xi_{\mathcal{K}}$ by interpreting each fresh predicate occurring in $\Xi_{\mathcal{K}}$ as follows.

$$
\begin{aligned}
(\mathbb{C})^{\mathcal{J}} &= \{a^{\mathcal{J}} \in \Delta^{\mathcal{I}} \mid a \in \mathsf{ind}_{\mathcal{K}}\} && \\
(\mathbb{S}_R)^{\mathcal{J}} &= \{o \in \Delta^{\mathcal{I}} \mid \langle o, o \rangle \in R^{\mathcal{I}}\} && \forall R \in \mathsf{rol}_{\mathcal{K}} \\
(\mathbb{D}_R)^{\mathcal{J}} &= (R)^{\mathcal{I}} && \forall R \in \mathsf{rol}_{\mathcal{K}}
\end{aligned}
$$

Because $\mathcal{I}$ is a description logic model of $\mathcal{K}$, interpretation $\mathcal{J}$ satisfies each rule in $\Xi_{\mathcal{T},\mathcal{R}}$ and each rule of the form (5.5)–(5.8) in $\mathsf{cl}_{\mathcal{K}}$. By the definition of $(\mathbb{C})^{\mathcal{J}}$ and $(\mathbb{S}_R)^{\mathcal{J}}$, interpretation

$\mathcal{J}$ also satisfies each rule of the form (5.1) and (5.2). Finally, $\mathcal{J}$ also satisfies each rule of the form (5.3) and (5.4) in $\mathsf{cl}_{\mathcal{K}}$, because for each role inclusion $S \sqsubseteq R \in \mathcal{R}$, we have that $S^{\mathcal{I}} \subseteq R^{\mathcal{I}}$; so $(\mathbb{S}_S)^{\mathcal{J}} \subseteq (\mathbb{S}_R)^{\mathcal{J}}$ and $(\mathbb{D}_S)^{\mathcal{J}} \subseteq (\mathbb{D}_R)^{\mathcal{J}}$. Therefore, $\mathcal{J}$ is a first-order model of $\Xi_{\mathcal{K}}$ such that $(\bot_c)^{\mathcal{J}} = \emptyset$ and $\mathcal{J} \not\models \pi(q)$, as required.

($\Leftarrow$) Assume that $\Xi_{\mathcal{K}} \not\models \exists y.\, \bot_c(y)$ and $\Xi_{\mathcal{K}} \not\models_{\mathcal{DL}} \pi(q)$; we show that $\mathcal{K} \not\models_{\mathcal{DL}} \pi(q)$. Recall that I is the universal interpretation of $\Xi_{\mathcal{K}}$; then I satisfies the following properties.

(a) By Theorem 2.3, for each term w, we have that $\bot_c(w) \notin \mathsf{inst}_I$, and $\|\pi(q)\|_I \not\subseteq \mathsf{inst}_I$.

(b) Due to rules (5.5)–(5.8) in $\mathsf{cl}_{\mathcal{K}}$, for all terms w and w' occurring in inst_I, we have that $\{\top_c(w), \top_r(w, w')\} \subseteq \mathsf{inst}_I$ and $\bot_r(w, w') \notin \mathsf{inst}_I$.

(c) By our translation of axioms of types (6) and (9) and due to rule (5.2) in $\mathsf{cl}_{\mathcal{K}}$, for each simple role $R \in \mathsf{rol}_{\mathcal{K}}$ and each term w, we have that $\mathbb{S}_R(w) \in \mathsf{inst}_I$ if and only if $R(w, w) \in \mathsf{inst}_I$.

We next associate with I the first-order interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ contains each term w occurring in inst_I and $\cdot^{\mathcal{I}}$ is specified as follows.

- For each individual $a \in \mathsf{ind}_{\mathcal{K}}$, let $a^{\mathcal{I}} = \|a\|_I$; for each individual $a \in (N_{\mathfrak{I}} \setminus \mathsf{ind}_{\mathcal{K}})$, let $a^{\mathcal{I}}$ be an arbitrary term in $\Delta^{\mathcal{I}}$.
- For each atomic concept $A \in N_{\mathfrak{C}}$, let $A^{\mathcal{I}} = \{w \in \Delta^{\mathcal{I}} \mid A(w) \in \mathsf{inst}_I\}$.
- For each role $R \in N_{\mathfrak{R}}$, let $R^{\mathcal{I}} = \{\langle w, w' \rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid R(w, w') \in \mathsf{inst}_I\}$.

Since $\mathcal{K}$ is normalised, the ABox $\mathcal{A}$ is non-empty; hence, $\Delta^{\mathcal{I}}$ is non-empty as well. By properties (a) and (b), we have that $\mathcal{I}$ is a description logic interpretation and $\mathcal{I} \not\models \pi(q)$. Clearly, interpretation $\mathcal{I}$ satisfies each assertion of $\mathcal{K}$ and each axiom of type other (7) of $\mathcal{K}$. By property (c), interpretation $\mathcal{I}$ also satisfies each axiom of type (7). Hence, $\mathcal{I}$ is a description logic model of $\mathcal{K}$ such that $\mathcal{I} \not\models \pi(q)$; that is, $\mathcal{K} \not\models_{\mathcal{DL}} \pi(q)$. □

5.3.2 Structural Properties of the Universal Interpretations of $\mathcal{K}$

We next prove the structural properties of the universal interpretation I of $\mathcal{K}$.

Lemma 5.3. *Interpretation I satisfies the following properties for all terms w and w' occurring in I and each role $R \in \mathsf{rol}_{\mathcal{K}}$.*

(1) Relation dir_I is a forest rooted in $N_{\mathfrak{I}}$.

(2) $R(w, w') \in \mathsf{inst}_I$ implies that a non-empty role chain $\rho = \chi_0 \cdot R_1 \cdot \chi_1 \cdots R_m \cdot \chi_m$ with $\rho \in \mathcal{L}(R)$ and terms $w_0, \ldots, w_m$ with $w_0 = w$ and $w_m = w'$ exist such that

(a) for each $i \in [1..m]$, either $w_i \in N_{\mathfrak{I}}$, or w_i is a labelled null of type R_i, A_i and $\mathbb{D}_{R_i}(w_{i-1}, w_i) \in \mathsf{inst}_I$.

(b) for each $i \in [1..m]$, we have that $R_i(w_{i-1}, w_i) \in \mathsf{inst}_I$; and

(c) for each $i \in [0..m]$ and each role S occurring in role chain χ_i, we have that $\{S(w_i, w_i), \mathbb{S}_S(w_i)\} \subseteq \mathsf{inst}_I$.

(3) $R(w, w) \in \mathsf{inst}_I$, R is simple, and $w \in \Phi_N$ imply that $\mathbb{S}_R(w) \in \mathsf{inst}_I$.

(4) $R(w, w') \in \mathsf{inst}_I$, R is simple, $w' \in \Phi_N$, and $w \neq w'$ imply that $\langle w, w' \rangle \in \mathsf{dir}_I$.

Proof. We first prove properties (1) and (2), later we prove properties (3) and (4).

Properties (1) and (2). Let I_0 be the instance that contains each ground atom occurring in $\Xi_{\mathcal{K}}$, and let $\langle I_1, r_1, \sigma_1 \rangle, \langle I_2, r_2, \sigma_2 \rangle \ldots$, be the chase sequence used to construct I. In the following, we assume w.l.o.g. that, for each $\langle I_n, r_n, \sigma_n \rangle$ where $r_n = \top_c(x) \wedge \top_c(y) \rightarrow \top_r(x, y)$ and $\sigma_n(y)$ is a labelled null, we have that $\top_r(\sigma_n(x), w_p) \in I_{n-1}$ for each edge $\langle w_p, \sigma_n(y) \rangle$ in $\mathsf{dir}_{I_{n-1}}$. Note that each chase sequence can be extended to a sequence that satisfies this property: it suffices to ensure that the sequence is saturated by rule $\top_c(x) \wedge \top_c(y) \rightarrow \top_r(x, y)$ before applying an existential rule of the form $A_1(x) \rightarrow \exists z. R(x, z) \wedge \mathbb{D}_R(x, z) \wedge A(z)$.

We prove by induction on $n \in \mathbb{N}$ that each instance I_n satisfies properties (1) and (2).

Base case. Consider instance I_0. Then inst_{I_0} does not contain labelled nulls, and so $\mathsf{dir}_{I_0} = \emptyset$ and property (1) vacuously holds. Consider an arbitrary atom $R(w, w') \in \mathsf{inst}_{I_0}$. Then, $\{w, w'\} \subseteq N_{\mathfrak{I}}$, and property (2) holds for $w_0 = w$, $w_1 = w'$, $\chi_0 = \epsilon$, and $\rho = \chi_0 \cdot R$.

Inductive step. Consider an arbitrary $n \in \mathbb{N}$ and assume that I_n satisfies properties (1) and (2). Then, instance I_{n+1} is the result of applying trigger $\langle r_{n+1}, \sigma_{n+1} \rangle$ to I_n. Next, by

considering each rule $r \in \Xi_{\mathcal{K}}$ that derives binary atoms and each ground substitution σ, we assume that $\langle r, \sigma \rangle = \langle r_{n+1}, \sigma_{n+1} \rangle$ and and we show that properties (1) and (2) hold for all fresh atoms in I_{n+1}.

Axioms of Type (3). Consider a rule r of the form $A(x) \rightarrow x \approx a$ in $\Xi_{\mathcal{K}}$ and consider an arbitrary term w_1. Let σ be the substitution with $\sigma(x) = w_1$, and assume that I_{n+1} is the result of applying $\langle r, \sigma \rangle$ to I_n. Let w_r and w'_r be the $<$-larger and $<$-smaller terms between w_1 and $\|a\|_{I_n}$. By the definition of $<$ and due to $\|a\|_{I_n} \in N_{\mathfrak{I}}$, we have that $w'_r \in N_{\mathfrak{I}}$. Then, instance $\mathsf{inst}_{I_{n+1}}$ is the result of replacing each occurrence of w_r with w'_r. We next consider the various atoms that get replaced by the application of $\langle r, \sigma \rangle$ to I_n and show that the properties hold.

- $R(w, w') \in I_n$ and either $w = w_r$ or $w' = w_r$. By the inductive hypothesis, there exist a non-empty role chain $\rho = \chi_0 \cdot R_1 \cdot \chi_1 \cdots R_m \cdot \chi_m$ with $\rho \in \mathcal{L}(R)$, and terms $w_0, \ldots, w_m$ with $w_0 = w$ and $w_m = w'$ that satisfy property (2). Then, property (2) holds for role chain ρ and terms $\|w_0\|_{I_{n+1}}, \ldots, \|w_m\|_{I_{n+1}}$.

- $\mathbb{D}_R(w, w_r) \in I_n$. Since $w'_r \in N_{\mathfrak{I}}$, we have that $\langle w, w'_r \rangle \notin \mathsf{dir}_{I_{n+1}}$, and property (1) vacuously holds.

- $\mathbb{D}_R(w_r, w') \in I_n$ and w' is a labelled null of type R, A. Hence, we have $\langle w_r, w' \rangle \in \mathsf{dir}_{I_n}$. By the inductive hypothesis, we have that $w_r \neq w'$ and w_r is the unique term in I_n such that $\langle w_r, w' \rangle \in \mathsf{dir}_{I_n}$. Since $w'_r \in N_{\mathfrak{I}}$, we have $\mathsf{dir}_{I_{n+1}}$ is a forest rooted in $N_{\mathfrak{I}}$.

Axioms of Type (8). Consider a rule r in $\Xi_{\mathcal{K}}$ of the following form.

$$r = A(x) \wedge A(y) \wedge \mathbb{C}(x) \wedge \mathbb{C}(y) \wedge \bigwedge_{1 \leq i \leq k} \mathbb{C}(z_i) \wedge R_i(x, z_i) \wedge R_i(y, z_i) \rightarrow x \approx y$$

Furthermore, consider arbitrary terms $w_x, w_y, w_1, \ldots, w_k$ occurring in I and let σ be the following substitution.

$$\sigma(x) = w_x \qquad \sigma(y) = w_y \qquad \sigma(z_i) = w_i \qquad 1 \leq i \leq k$$

Assume that I_{n+1} is the result of applying $\langle r, \sigma \rangle$ to I_n. Let w_r and w'_r be the $<$-larger and

$<$-smaller terms between w_x and w_y. Since $\{\mathbb{C}(x), \mathbb{C}(y)\} \subseteq \varphi$, we have that w_r and w'_r are individuals from $N_\mathfrak{J}$. Then, instance $\mathsf{inst}_{I_{n+1}}$ is the result of replacing each occurrence of w_r with w'_r. Similarly as for the case of axioms of type (3), one can show that the properties are preserved for all atoms in I_n that are replaced by applying $\langle r, \sigma \rangle$ to I_n.

Axioms of Type (4). Consider a rule r of the form $A_1(x) \rightarrow \exists z.R(x,z) \wedge \mathbb{D}_R(x,z) \wedge A(z)$ in $\Xi_\mathcal{K}$ and an arbitrary term w. Let σ be the substitution with $\sigma(x) = w$, and assume that I_{n+1} is the result of applying $\langle r, \sigma \rangle$ to I_n. Then, we have that instance $\mathsf{inst}_{I_{n+1}}$ extends inst_{I_n} with $\{R(w,w'), \mathbb{D}_R(w,w'), A(w')\}$, where w' is the $<$-smallest labelled null not occurring in I_n and the type of w' is R, A. By the inductive hypothesis, we have that dir_{I_n} is a forest rooted in $N_\mathfrak{J}$. Since w' is fresh and $\mathsf{dir}_{I_{n+1}}$ extends dir_{I_n} with $\langle w, w' \rangle$, property (1) holds. Moreover, property (2) holds for $w_0 = w$, $w_1 = w'$, $\chi_0 = \epsilon$, and $\rho = \chi_0 \cdot R$.

Axioms of Type (6) and (9). Consider a rule r of the form $A_1(x) \rightarrow R(x,x) \wedge \mathbb{S}_R(x)$ in $\Xi_\mathcal{K}$, and consider an arbitrary term w. Let σ be the substitution such that $\sigma(x) = w$, and assume that I_{n+1} is the result of applying $\langle r, \sigma \rangle$ to I_n. Then, $\mathsf{inst}_{I_{n+1}}$ extends inst_{I_n} with $\{R(w,w), \mathbb{S}_R(w)\}$. Property (1) holds by the inductive hypothesis, and property (2) holds for $w_0 = w$, $\chi_0 = R$, and $\rho = \chi_0$.

Axioms of Type (10). Consider a rule r of the form $\bigwedge_{i=1}^{k} R_i(x_{i-1}, x_i) \rightarrow R(x_0, x_k)$ in $\Xi_\mathcal{K}$ and consider arbitrary terms $w_0, \ldots, w_k$. Let σ be the substitution such that $\sigma(x_i) = w_i$ for each $i \in [0..k]$, and assume that I_{n+1} is the result of applying $\langle r, \sigma \rangle$ to I_n. Then, for each $i \in [1..k]$, we have that $R_i(w_{i-1}, w_i) \in \mathsf{inst}_{I_n}$, and $\mathsf{inst}_{I_{n+1}}$ extends inst_{I_n} with $R(w_0, w_k)$. Then, property (1) holds by the inductive hypothesis. For property (2), by the inductive hypothesis, for each $i \in [1..k]$, terms $w^i_0, \ldots, w^i_{m_i}$ with $w^i_0 = w_{i-1}$ and $w^i_{m_i} = w_i$ and a non-empty role chain ρ^i with $\rho^i \in \mathcal{L}(R_i)$ exist that satisfy property (2); note that $w^1_0 = w_0 = w$, that $w^k_{m_k} = w_m = w'$, and that $w^i_{m_i} = w^{i+1}_0$ for each $i \in [1..k-1]$. By the definition of $\mathcal{L}(R)$, then $\rho^1 \cdots \rho^k \in \mathcal{L}(R)$, and so property (2) holds for role chain $\rho^1 \cdots \rho^k$ and terms $w_0, w^1_1, \ldots, w^1_{m_1}, \ldots, w^k_1, \ldots, w^k_{m_k}$.

Rules of the form (5.4). Consider a rule r of the form $\mathbb{D}_S(x,y) \rightarrow \mathbb{D}_R(x,y)$ in $\Xi_\mathcal{K}$ and consider arbitrary terms w and w'. Let σ be the substitution such that $\sigma(x) = w$ and $\sigma(y) = w'$, and assume that I_{n+1} is the result of applying $\langle r, \sigma \rangle$ to I_n. Then, property (1)

and (2) hold by the inductive hypothesis.

Rule (5.7). Consider the rule $r = \top_c(x) \wedge \top_c(y) \rightarrow \top_r(x, y)$ in $\Xi_{\mathcal{K}}$, and consider arbitrary terms w and w'. Let σ be the substitution with $\sigma(x) = w$ and let $\sigma(y) = w'$, and assume that I_{n+1} is the result of applying $\langle r, \sigma \rangle$ to I_n. Then, $\mathsf{inst}_{I_{n+1}}$ extends inst_{I_n} with $\top_r(w, w')$. Property (1) holds by the inductive hypothesis. For property (2), in case $w' \in N_{\mathfrak{I}}$, the property holds for $w_0 = w$, $w_1 = w'$, $\chi_0 = \epsilon$, and $\rho = \chi_0 \cdot \top_r$. Hence, in the following, we consider the case in which $w' \in \Phi_N$. By the form of the existential rules occurring in $\Xi_{\mathcal{K}}$, term w' must have been generated by the application of a rule of the form $A_1(x) \rightarrow \exists z.R(x, z) \wedge \mathbb{D}_R(x, z) \wedge A(z)$; thus, a term w_p exists such that $\{R(w_p, w'), \mathbb{D}_R(w_p, w')\} \subseteq \mathsf{inst}_{I_n}$. By the initial assumption on the chase sequence, we have $\top_r(w, w_p) \in \mathsf{inst}_{I_n}$. By the inductive hypothesis, a non-empty role chain ρ with $\rho \in \mathcal{L}(\top_r)$ and terms $w_0, \ldots, w_m$ with $w_0 = w$ and $w_m = w_p$ exist that satisfy property (2). By the definition of $\mathcal{L}(\top_r)$, we then have that $\rho \cdot R \in \mathcal{L}(\top_r)$. Since $R(w_p, w') \in \mathsf{inst}_{I_n}$, property (2) holds for role chain $\rho \cdot R$ and terms $w_0, \ldots, w_m, w'$.

Property (3). Consider an arbitrary role $R \in \mathsf{rol}_{\mathcal{K}}$ that is simple in $\mathcal{R}$ and an arbitrary labelled null $w \in \Phi_N$. Assume that $R(w, w) \in \mathsf{inst}_I$; we show that $\mathbb{S}_R(w) \in \mathsf{inst}_I$. By property (2), a non-empty role chain $\rho = \chi_0 \cdot R_1 \cdot \chi_1 \cdots R_m \cdot \chi_m$ with $\rho \in \mathcal{L}(R)$ and terms $w_0, \ldots, w_m$ with $w_0 = w$ and $w_m = w$ exist that satisfy the conditions in property (2). Moreover, because R is simple and ρ is non-empty, we have $|\rho| = 1$. We distinguish two cases.

- $\chi_0 = \emptyset$ and $\rho = R_1$. Since $w \in \Phi_N$, we have that $\mathbb{D}_{R_1}(w, w) \in \mathsf{inst}_I$. Hence, we have that $\langle w, w \rangle \in \mathsf{dir}_I$, which contradicts property (1).
- $\chi_0 = S$ and $\rho = \chi_0$. Then, we have $S \in \mathcal{L}(R)$ and $\mathbb{S}_S(w) \in \mathsf{inst}_I$. Since $S \in \mathcal{L}(R)$ and R is simple in $\mathcal{R}$, we have that roles $R_0, \ldots, R_k$ with $R_0 = S$ and $R_k = R$ exist such that $R_{i-1} \sqsubseteq R_i \in \mathcal{R}$ for each $i \in [1..k]$. Due to rules of the form (5.3) in $\Xi_{\mathcal{K}}$ and because I is closed under $\Xi_{\mathcal{K}}$, we have $\mathbb{S}_R(w) \in \mathsf{inst}_I$ and property (3) holds.

Property (4). Consider an arbitrary role $R \in \mathsf{rol}_{\mathcal{K}}$ that is simple in $\mathcal{R}$, and consider arbitrary terms w and w' such that $w \neq w'$ and $w' \in \Phi_N$. Assume that $R(w, w') \in \mathsf{inst}_I$, we

show that $\langle w, w\rangle \in \mathsf{dir}_I$. By property (2), a non-empty role chain $\rho = \chi_0 \cdot R_1 \cdot \chi_1 \cdots R_m \cdot \chi_m$ with $\rho \in \mathcal{L}(R)$ and terms $w_0, \ldots, w_m$ with $w_0 = w$ and $w_m = w'$ exist that satisfy all the conditions in (2). Because R is a simple role, we have $|\rho| = 1$; moreover, due to $w \neq w'$, we also have that $\chi_0 = \emptyset$ and ρ is of the form $\rho = R_1$. Since $\mathbb{D}_{R_1}(w, w') \in \mathsf{inst}_I$ and $w' \in \Phi_N$, we have that $\langle w, w'\rangle \in \mathsf{dir}_I$ and property (4) holds. □

5.3.3 Correctness of the Translation into Datalog

Finally, we prove Proposition 5.7. Towards this goal, we show that, for each atomic concept $A \in \mathsf{con}_\mathcal{K}$ and each individual $a \in \mathsf{ind}_\mathcal{K}$, we have that

(a) $\|A(a)\|_I \in \mathsf{inst}_I$ if and only if $\|A(a)\|_J \in \mathsf{inst}_J$.

Please note that, by Proposition 5.2 and Theorem 2.3, property (a) implies Proposition 5.7.

We next define a function ι that maps each term w occurring in I to a term $\iota(w)$ as follows.

$$\iota(w) = \begin{cases} w & \text{if } w \in N_\mathfrak{I}, \\ o_{R,A} & \text{if } w \in \Phi_N \text{ and the type of } w \text{ is } R, A. \end{cases}$$

Lemma 5.9 shows that ι is an homomorphism from inst_I to inst_J modulo the replacements of individuals due to equality rules, and that whenever a term w is replaced by w' in I, then $\iota(w)$ and $\iota(w')$ are represented by the same canonical representative in J.

Lemma 5.9. *Function ι satisfies the following three properties for all terms w and w', each $A \in \{\mathbb{C}\} \cup \mathsf{con}_\mathcal{K} \cup \{\mathbb{S}_S \mid S \in \mathsf{rol}_\mathcal{K}\}$, and each $R \in \mathsf{rol}_\mathcal{K} \cup \{\mathbb{D}_S \mid S \in \mathsf{rol}_\mathcal{K}\}$.*

(1) $A(w) \in \mathsf{inst}_I$ implies that $\|A(\iota(w))\|_J \in \mathsf{inst}_J$.

(2) $R(w, w') \in \mathsf{inst}_I$ implies that $\|R(\iota(w), \iota(w'))\|_J \in \mathsf{inst}_J$.

(3) $w \rightsquigarrow w' \in \mathsf{eq}_I$ implies that $\|\iota(w)\|_J = \|\iota(w')\|_J$.

Proof. Let $\langle I_1, r_1, \sigma_1\rangle, \langle I_2, r_2, \sigma_2\rangle, \ldots$ be the chase sequence used to construct I and let I_0 be the instance that contains each ground atom in $\Xi_\mathcal{K}$. We show by induction on $n \in \mathbb{N}$ that each instance I_n satisfies these three properties.

Base case. Consider instance I_0. Then, inst_{I_0} contains precisely the atoms occurring in $\Xi_{\mathcal{K}}$, whereas $\mathsf{eq}_{I_0} = \emptyset$. Hence, property (3) holds vacuously; furthermore, for each term w occurring in I_0, we have that $w \in N_{\mathfrak{I}}$ and $\iota(w) = w$. For properties (1) and (2), consider an arbitrary atom $\phi \in \mathsf{inst}_{I_0}$. By the definition of $\mathsf{D}_{\mathcal{K}}$, we have that $\phi \in \mathsf{D}_{\mathcal{K}}$ and so $\mathsf{D}_{\mathcal{K}} \models \phi$. By Theorem 2.3, we then have that $\|\phi\|_J \in \mathsf{inst}_J$, as required.

Inductive step. Consider an arbitrary $n \in \mathbb{N}$ and assume that the instance I_n satisfies properties (1)–(3).

We first prove that I_n satisfies the following auxiliary property for all terms w and w' occurring in I_n.

(4) $\|w\|_{I_n} = w'$ implies that $\|\iota(w)\|_J = \|\iota(w')\|_J$.

Consider arbitrary terms w and w' occurring in I_n, and assume that $\|w\|_{I_n} = w'$. Then, terms $w_0, \ldots, w_k$ with $w_0 = w$ and $w_k = w'$ exist such that $w_{i-1} \rightsquigarrow w_i \in \mathsf{eq}_{I_n}$ for each $i \in [1..k]$. Since I_n satisfies property (3) by the inductive hypothesis, for each $i \in [1..k]$, we have $\|\iota(w_{i-1})\|_J = \|\iota(w_i)\|_J$; therefore, we have $\|\iota(w)\|_J = \|\iota(w')\|_J$.

Instance I_{n+1} is the result of applying trigger $\langle r_{n+1}, \sigma_{n+1} \rangle$ to I_n. Next, by considering each rule $r \in \Xi_{\mathcal{K}}$ and each ground substitution σ, we assume that $\langle r, \sigma \rangle = \langle r_{n+1}, \sigma_{n+1} \rangle$ and show that properties (1)–(3) hold for all fresh atoms in I_{n+1}.

(Datalog Rule) Consider a rule r of the form $\varphi(\vec{x}, \vec{y}) \to \psi(\vec{x})$ in $\Xi_{\mathcal{K}}$ and a substitution σ with $\mathsf{dom}(\sigma) = \vec{x} \cup \vec{y}$, and assume that I_{n+1} is the result of applying $\langle r, \sigma \rangle$ to I_n. Then, we have $\sigma(\|\varphi\|_{I_n}) \subseteq \mathsf{inst}_{I_n}$, and $\mathsf{inst}_{I_{n+1}} = \mathsf{inst}_{I_n} \cup \sigma(\|\psi\|_{I_n})$ and $\mathsf{eq}_{I_{n+1}} = \mathsf{eq}_{I_n}$. Let σ' be the substitution with $\sigma'(z) = \iota(\sigma(z))$ for each $z \in \vec{x} \cup \vec{y}$. By the inductive hypothesis, we have $\|\sigma'(\varphi)\|_J \subseteq \mathsf{inst}_J$. Since $r \in \mathsf{D}_{\mathcal{K}}$ and J is closed under $\mathsf{D}_{\mathcal{K}}$, we have that $\|\sigma'(\psi)\|_J \subseteq \mathsf{inst}_J$.

(Existential Rule) Consider a rule r of the form $A_1(x) \to \exists z.R(x, z) \wedge \mathbb{D}_R(x, z) \wedge A(z)$ in $\Xi_{\mathcal{K}}$ and a substitution $\sigma = \{x \mapsto w\}$, and assume that I_{n+1} is the result of applying $\langle r, \sigma \rangle$ to I_n. Then, we have $A(w) \in \mathsf{inst}_{I_n}$, and $\mathsf{inst}_{I_{n+1}} = \mathsf{inst}_{I_n} \cup \{R(w, w'), \mathbb{D}_R(w, w'), A(w')\}$ and $\mathsf{eq}_{I_{n+1}} = \mathsf{eq}_{I_n}$, where $w' \in \Phi_N$ is the $<$-smallest labelled null not occurring in I_n. Then w' is generated by rule r and $\iota(w') = o_{R,A}$. By the inductive hypothesis, we have $\|A_1(\iota(w))\|_J \in J$. Since $\mathsf{D}_{\mathcal{K}}$ contains the rule $A_1(x) \to R(x, o_{R,A}) \wedge \mathbb{D}_R(x, o_{R,A}) \wedge A(o_{R,A})$

and J is closed under $\mathsf{D}_{\mathcal{K}}$, $\|R(\iota(w), \iota(w')) \wedge \mathbb{D}_R(\iota(w), \iota(w')) \wedge A(\iota(w'))\|_J \subseteq J$, as required.

(Equality Rule) We distinguish between two types of equality rules.

First, consider an equality rule r of the form $A(x) \to x \approx a$ in $\Xi_{\mathcal{K}}$ and a substitution $\sigma = \{x \mapsto w_1\}$, and assume that I_{n+1} is the result of applying $\langle r, \sigma \rangle$ to I_n. Then, we have $A(w_1) \in \mathsf{inst}_{I_n}$ and $w_1 \neq \|a\|_{I_n}$. By the inductive hypothesis, we have $\|A(\iota(w_1))\|_J \in J$. Since $r \in \mathsf{D}_{\mathcal{K}}$ and J is closed under $\mathsf{D}_{\mathcal{K}}$, we have $\|\iota(w_1)\|_J = \|a\|_J$. We next consider two cases, and in each case we define terms w and w' such that $w > w'$ and $\|\iota(w)\|_J = \|\iota(w')\|_J$.

- $w_1 > \|a\|_{I_n}$. Let $w = w_1$ and let $w' = \|a\|_{I_n}$ Due to $w' = \|a\|_{I_n}$ and property (4), we have $\|\iota(w')\|_J = \|a\|_J$. Since $\|\iota(w_1)\|_J = \|a\|_J$, we have $\|\iota(w)\|_J = \|\iota(w')\|_J$.
- $\|a\|_{I_n} > w_1$. Let $w = \|a\|_{I_n}$ and let $w' = w_1$. Due to $w = \|a\|_{I_n}$ and property (4), we have $\|\iota(w)\|_J = \|a\|_J$. Since $\|\iota(w_1)\|_J = \|a\|_J$, we have $\|\iota(w')\|_J = \|\iota(w)\|_J$.

In either cases, we then have $\|\iota(w)\|_J = \|\iota(w')\|_J$. Hence, the instance I_{n+1}—obtained from I_n by replacing each occurrence of term w in inst_{I_n} with w', and by extending eq_{I_n} with $w \rightsquigarrow w'$—satisfies properties (1)–(3).

Second, consider a rule $r = \varphi \to x \approx y$ in $\Xi_{\mathcal{K}}$, where φ is of the form

$$\varphi = A(x) \wedge A(y) \wedge \mathbb{C}(x) \wedge \mathbb{C}(y) \wedge \bigwedge_{1 \leq i \leq k} \mathbb{C}(z_i) \wedge R_i(x, z_i) \wedge R_i(y, z_i).$$

Moreover, consider a ground substitution σ, and assume that I_{n+1} is the result of applying $\langle r, \sigma \rangle$ to I_n. Then, we have $\sigma(\varphi) \subseteq \mathsf{inst}_{I_n}$ and $\sigma(x) \neq \sigma(y)$. Let σ' be the substitution such that $\sigma'(z) = \iota(\sigma(z))$ for each variable z occurring in φ. By the inductive hypothesis, we have $\|\sigma'(\varphi)\|_J \subseteq \mathsf{inst}_J$. Since $r \in \mathsf{D}_{\mathcal{K}}$ and J is closed under $\mathsf{D}_{\mathcal{K}}$, we have $\|\sigma'(x)\|_J = \|\sigma'(y)\|_J$. Let w and w' be the $<$-larger and $<$-smaller terms in $\{\sigma(x), \sigma(y)\}$, respectively. Due to $\|\sigma'(x)\|_J = \|\sigma'(y)\|_J$, we also have that $\|\iota(w)\|_J = \|\iota(w')\|_J$; so the instance I_{n+1}—obtained from I_n by replacing each occurrence of term w in inst_{I_n} with w', and by extending eq_{I_n} with $w \rightsquigarrow w'$—satisfies properties (1)–(3). □

In Lemma 5.10 we prove a similar result for the other direction—that is, we show how ι can be used to map atoms in inst_J to atoms in inst_I modulo renaming of terms—in

particular, property (5) shows that each direct edge in J corresponds to many direct edges in I all of which are labelled by the same direct predicates. Please recall that, given two constants u and u', we say that u is directly connected to u' in J if a role S exists such that $\mathbb{D}_S(u, u') \in \mathsf{inst}_J$.

Lemma 5.10. *Function ι satisfies the following properties for all terms w and w' occurring in I, each constant $o_{P,B} \in N_{aux}$, each concept $A \in \{\mathbb{C}\} \cup \mathsf{con}_{\mathcal{K}}$, and each role $R \in \mathsf{rol}_{\mathcal{K}}$.*

(1) $A(\iota(w)) \in \mathsf{inst}_J$ *implies that* $\|A(w)\|_I \in \mathsf{inst}_I$.

(2) $\mathbb{S}_R(\iota(w)) \in \mathsf{inst}_J$ *implies that* $\|\mathbb{S}_R(w) \wedge R(w, w)\|_I \in \mathsf{inst}_I$.

(3) $R(\iota(w), \iota(w')) \in \mathsf{inst}_J$, *and* $\top_r \in \mathcal{L}(R)$ *or* $\iota(w') \in N_{\mathfrak{I}}$ *imply that* $\|R(w, w')\|_I \in \mathsf{inst}_I$.

(4) $R(\iota(w), \iota(w')) \in \mathsf{inst}_J$, $\top_r \notin \mathcal{L}(R)$, *and* $\iota(w') \notin N_{\mathfrak{I}}$ *imply that a term w'' occurring in I exists such that* $\iota(w'') = \iota(w')$ *and* $\|R(w, w'')\|_I \in \mathsf{inst}_I$.

(5) $\iota(w)$ *is directly connected to* $o_{P,B}$ *in* J *implies that a term* w^* *exists in* I *such that*

- $\iota(w^*) = o_{P,B}$ *and* $\|P(w, w^*) \wedge \mathbb{D}_P(w, w^*)\|_I \subseteq \mathsf{inst}_I$, *and*
- *for each role* $S \in \mathsf{rol}_{\mathcal{K}}$, $\mathbb{D}_S(\iota(w), o_{P,B}) \in \mathsf{inst}_J$ *implies that* $P \sqsubseteq^*_{\mathcal{R}} S$.

(6) For each constant u occurring in J, a term w_u exists such that $\iota(w_u) = u$.

(7) $\iota(w) \rightsquigarrow \iota(w') \in \mathsf{eq}_J$ *implies that* $\|w\|_I = \|w'\|_I$.

Proof. Let $\langle J_1, r_1, \sigma_1 \rangle, \langle J_2, r_2, \sigma_2 \rangle, \ldots$ be the chase sequence for $\mathsf{D}_{\mathcal{K}}$ used to construct J and let J_0 be the instance that contains all ground atoms in $\mathsf{D}_{\mathcal{K}}$. Please note that, by the definition of $\mathsf{cl}_{\mathcal{K}}$ and the form of equality rules in $\mathsf{D}_{\mathcal{K}}$, for each $n \in \mathbb{N}$ and all constants u and u' occurring in J_n, we have that

- $\mathbb{C}(u) \in \mathsf{inst}_{J_n}$ implies that $u \in N_{\mathfrak{I}}$, and
- $u \rightsquigarrow u' \in \mathsf{eq}_{J_n}$ implies that $u' \in N_{\mathfrak{I}}$.

Next, we prove by induction on $n \in \mathbb{N}$ that each instance J_n satisfies properties (1)–(7).

Base case. Consider instance J_0. Then, inst_{J_0} contains precisely the ground atoms occurring in $\mathsf{D}_{\mathcal{K}}$, whereas $\mathsf{eq}_{J_0} = \emptyset$. Moreover, all atoms in inst_{J_0} are constructed using the

individuals from $\mathsf{ind}_{\mathcal{K}}$ and the predicates in $\{\mathbb{C}\} \cup \mathsf{con}_{\mathcal{K}} \cup \mathsf{rol}_{\mathcal{K}}$. Thus, for each constant u occurring in J_0, we have that $\iota(u) = u$; furthermore, properties (2) and (4)–(7) hold vacuously. For the remaining properties, consider an arbitrary atom $\phi \in J_0$. By the definition of $\mathsf{D}_{\mathcal{K}}$, we have $\phi \in \Xi_{\mathcal{K}}$ and so $\Xi_{\mathcal{K}} \models \phi$. By Theorem 2.3, we then have that $\|\phi\|_I \in \mathsf{inst}_I$.

Inductive step. Consider an arbitrary $n \in \mathbb{N}$ and assume that J_n satisfies (1)–(7).

We first show that J_n satisfies the following auxiliary property for each term w and each individual a occurring in I:

(8) $\|\iota(w)\|_{J_n} = a$ implies that $\|w\|_I = \|a\|_I$.

Consider an arbitrary term w and an arbitrary individual a occurring in I, and assume that $\|\iota(w)\|_{J_n} = a$. Then, terms $w_0, \ldots, w_k$ with $w_0 = w$ and $w_k = a$ exist in I such that $\iota(w_{i-1}) \rightsquigarrow \iota(w_i) \in \mathsf{eq}_{J_n}$ for each $i \in [1..k]$. Since J_n satisfies property (7), by the inductive hypothesis, for each $i \in [1..k]$, we have that $\|w_{i-1}\|_I = \|w_i\|_I$; and so $\|w\|_I = \|a\|_I$.

Instance J_{n+1} is the result of applying trigger $\langle r_{n+1}, \sigma_{n+1} \rangle$ to J_n. Next, by considering each rule $r \in \mathsf{D}_{\mathcal{K}}$ and each ground substitution σ, we assume that $\langle r, \sigma \rangle = \langle r_{n+1}, \sigma_{n+1} \rangle$ and show that properties (1)–(7) hold for all fresh atoms in J_{n+1}.

Axioms of Type (1), (2), and (7) and Rule of the form (5.5). Consider a rule r of the form $\bigwedge_{i=1}^{k} A_i(x) \to A(x)$ in $\mathsf{D}_{\mathcal{K}}$ where each $A_{(i)} \in \{\mathbb{C}\} \cup \mathsf{con}_{\mathcal{K}}$, and consider an arbitrary term w occurring in I. Let σ be the substitution with $\sigma(x) = \iota(w)$ and assume that J_{n+1} is the result of applying $\langle r, \sigma \rangle$ to J_n. Then, for each $i \in [1..k]$, we have $A_i(\iota(w)) \in \mathsf{inst}_{J_n}$; furthermore, we have that $\mathsf{inst}_{J_{n+1}} = \mathsf{inst}_{J_n} \cup \{A(\iota(w))\}$ and $\mathsf{eq}_{J_{n+1}} = \mathsf{eq}_{J_n}$. By the inductive hypothesis, for each $i \in [1..k]$, we have $\|A_i(w)\|_I \in \mathsf{inst}_I$. As $r \in \Xi_{\mathcal{K}}$ and I is closed under $\Xi_{\mathcal{K}}$, we have $\|A(w)\|_I \in \mathsf{inst}_I$.

Axioms of Type (3). Consider a rule r of the form $A(x) \to x \approx a$ and an arbitrary term w_1 occurring in I. Let σ be the substitution with $\sigma(x) = \iota(w_1)$, and assume that J_{n+1} is the result of applying $\langle r, \sigma \rangle$ to J_n. Then, we have that $A(\iota(w_1)) \in \mathsf{inst}_{J_n}$ and $\iota(w_1) \neq \|a\|_{J_n}$. By the inductive hypothesis, we have $\|A(w_1)\|_I \in \mathsf{inst}_I$. Since $r \in \Xi_{\mathcal{K}}$ and I is closed under $\Xi_{\mathcal{K}}$, we have $\|w_1\|_I = \|a\|_I$. We next consider two cases, and in each case we define terms w and w' such that $\iota(w) > \iota(w')$, $\iota(w') \in N_{\mathfrak{I}}$, and $\|w\|_I = \|w'\|_I$.

- $\iota(w_1) > \|a\|_{J_n}$. Let $w = w_1$ and let $w' = \|a\|_{J_n}$. Then, we have that $w' \in N_\mathfrak{I}$, and so $\iota(w') = \|a\|_{J_n}$. By property (8), we then have $\|w'\|_I = \|a\|_I$. Since $\|w_1\|_I = \|a\|_I$ and $w = w_1$, we have $\|w\|_I = \|w'\|_I$.

- $\|a\|_{J_n} > \iota(w_1)$. Let $w = \|a\|_{J_n}$ and let $w' = w_1$. Then, we have $\iota(w') \in N_\mathfrak{I}$ and, due to $w = \|a\|_{J_n}$, we also have $w \in N_\mathfrak{I}$ and $\iota(w) = \|a\|_{J_n}$. By property (8), we then have $\|w\|_I = \|a\|_I$. Since $w_1 = w$ and $\|w_1\|_I = \|a\|_I$, we have $\|w'\|_I = \|w\|_I$.

Then, J_{n+1} is obtained from J_n by replacing each occurrence of $\iota(w)$ in inst_{J_n} with $\iota(w')$, and by extending eq_{J_n} with $\iota(w) \rightsquigarrow \iota(w')$. Due to $\|w\|_I = \|w'\|_I$, properties (6) and (7), as well as (1) and (2) hold. We next show that properties (3) and (4) hold for each fresh binary atom in J_{n+1}.

Consider an arbitrary atom $R(\iota(w_r), \iota(w'_r)) \in \mathsf{inst}_{J_n}$ that is replaced by the application of trigger $\langle r, \sigma \rangle$. In the following, we consider only the case in which $R \notin \mathcal{L}(\top_r)$ and $\iota(w'_r) \notin N_\mathfrak{I}$; in all other cases, properties (3) and (4) are preserved because J_{n+1} satisfies property (7). Then, one of the following must hold.

- $\iota(w_r) = \iota(w)$ and $\iota(w'_r) = \iota(w)$. By the inductive hypothesis, a term w'' exists such that $\iota(w'') = \iota(w'_r)$ and $\|R(w_r, w'')\|_I \in \mathsf{inst}_I$. Since $\iota(w_r) = \iota(w)$ and $\iota(w'') = \iota(w)$, by property (7) we have $\|w_r\|_I = \|w'\|_I$ and $\|w''\|_I = \|w'\|_I$; thus $\|R(w', w')\|_I \in \mathsf{inst}_I$.

- $\iota(w_r) = \iota(w)$ and $\iota(w'_r) \neq \iota(w)$. By the inductive hypothesis, a term w'' exists such that $\iota(w'') = \iota(w'_r)$ and $\|R(w_r, w'')\|_I \in \mathsf{inst}_I$. Since $\iota(w_r) = \iota(w)$, by property (7), we have that $\|w_r\|_I = \|w'\|_I$; thus $\|R(w', w'')\|_I \in \mathsf{inst}_I$.

- $\iota(w_r) \neq \iota(w)$ and $\iota(w'_r) = \iota(w)$. By the inductive hypothesis, a term w'' exists such that $\iota(w'') = \iota(w'_r)$ and $\|R(w_r, w'')\|_I \in \mathsf{inst}_I$. By property (7) and since $\iota(w'') = \iota(w)$, we have $\|w''\|_I = \|w'\|_I$; thus $\|R(w_r, w')\|_I \in \mathsf{inst}_I$.

For property (5), consider an arbitrary term w_r and an arbitrary auxiliary constant $o_{P,B}$ such that $\iota(w_r)$ is directly connected to $o_{P,B}$ in J_n. We distinguish two alternative cases.

- $\iota(w_r) = \iota(w)$ and $\iota(w) \neq o_{P,B}$. By the inductive hypothesis, a term w^* exists such that $\iota(w^*) = o_{P,B}$ and $\|P(w_r, w^*) \wedge \mathbb{D}_P(w_r, w^*)\|_I \subseteq \mathsf{inst}_I$. Since $\iota(w_r) = \iota(w)$, by property (7), we have $\|w_r\|_I = \|w'\|_I$, and so $\|P(w', w^*) \wedge \mathbb{D}_P(w', w^*)\|_I \subseteq \mathsf{inst}_I$.

- $\iota(w) = o_{P,B}$. Given that $\iota(w') \in N_{\mathfrak{I}}$, the property vacuously holds.

Axioms of Type (4). Consider a rule $r \in \mathsf{D}_{\mathcal{K}}$ of the form

$$r = A_1(x) \rightarrow R(x, o_{R,A}) \wedge \mathbb{D}_R(x, o_{R,A}) \wedge A(o_{R,A})$$

and consider an arbitrary term w occurring in I. Let σ be the substitution with $\sigma(x) = \iota(w)$, and assume that J_{n+1} is the result of applying $\langle r, \sigma \rangle$ to J_n. Then, we have $A_1(\iota(w)) \in \mathsf{inst}_{J_n}$, as well as $\mathsf{eq}_{J_{n+1}} = \mathsf{eq}_{J_n}$ and

$$\mathsf{inst}_{J_{n+1}} = \mathsf{inst}_{J_n} \cup \|R(\iota(w), o_{R,A}) \wedge \mathbb{D}_R(\iota(w), o_{R,A}) \wedge A(o_{R,A})\|_{J_n}.$$

By the inductive hypothesis, we have that $\|A_1(w)\|_I \in \mathsf{inst}_I$. Furthermore, $\Xi_{\mathcal{K}}$ contains a rule $A_1(x) \rightarrow \exists z.R(x, z) \wedge \mathbb{D}_R(x, z) \wedge A(z)$. Since I is closed under $\Xi_{\mathcal{K}}$, a term w^* exists in I such that $\iota(w^*) = o_{R,A}$ and $\|R(w, w^*) \wedge \mathbb{D}_R(w, w^*) \wedge A(w^*)\|_I \subseteq \mathsf{inst}_I$. We next show that the properties the are satisfied by considering two alternative cases.

- $\|o_{R,A}\|_{J_n} = o_{R,A}$. Then, due to $\iota(w^*) = o_{R,A}$, property (6) is satisfied; furthermore, as $R \sqsubseteq^*_{\mathcal{R}} R$, property (5) is also satisfied. Next, we consider an arbitrary term w' with $\iota(w') = o_{R,A}$ and show that properties (1) and (3)–(4) are satisfied by the fresh atoms in $\mathsf{inst}_{J_{n+1}}$. By the definition of $\Xi_{\mathcal{K}}$ and I, term w' must have been generated by a rule in $\Xi_{\mathcal{K}}$ of the form $A'_1(x) \rightarrow \exists z.R(x, z) \wedge \mathbb{D}_R(x, z) \wedge A(z)$; thus $\|A(w')\|_I \in \mathsf{inst}_I$ and property (1) holds. For properties (3) and (4), we consider two cases.
  - $\top_r \in \mathcal{L}(R)$. Since $\|A_1(w) \wedge A(w')\|_I \subseteq \mathsf{inst}_I$, $\Xi_{\mathcal{K}}$ contains rules (5.5) and (5.7), and I is closed under $\Xi_{\mathcal{K}}$, we have that $\|\top_r(w, w')\|_I \in \mathsf{inst}_I$. As $\top_r \in \mathcal{L}(R)$ and I is closed under $\Xi_{\mathcal{K}}$, we have that $\|R(w, w')\|_I \in \mathsf{inst}_I$ and property (3) holds.
  - $\top_r \notin \mathcal{L}(R)$. Then property (4) is satisfied by term $w'' = w^*$.
- $\|o_{R,A}\|_{J_n} \neq o_{R,A}$. Hence, auxiliary constant $o_{R,A}$ occurs already in J_n and property (6) holds by the inductive hypothesis. Next, we consider an arbitrary term w' with $\iota(w') = \|o_{R,A}\|_{J_n}$ and show that properties (1) and (3) hold. Then, we have that

$\iota(w') \in N_{\mathfrak{I}}$ and $\iota(w') = \|\iota(w^*)\|_{J_n}$. By property (8), we have $\|w'\|_I = \|w^*\|_I$; thus $\|R(w, w') \wedge A(w')\|_I \subseteq \mathsf{inst}_I$ and the properties hold.

Axioms of Type (5). Consider a rule r of the form $R(x, y) \wedge A_1(y) \rightarrow A(x)$ in $\mathsf{D}_{\mathcal{K}}$, and two arbitrary terms w and w' occurring in I. Let σ be the substitution with $\sigma(x) = \iota(w)$ and $\sigma(y) = \iota(w')$, and assume that J_{n+1} is the result of applying $\langle r, \sigma \rangle$ to J_n. Then, we have that $\{R(\iota(w), \iota(w')), A_1(\iota(w'))\} \subseteq \mathsf{inst}_{J_n}$, $\mathsf{eq}_{J_{n+1}} = \mathsf{eq}_{J_n}$, and $\mathsf{inst}_{J_{n+1}} = \mathsf{inst}_{J_n} \cup \{A(\iota(w)\}$. We distinguish two cases.

- $\top_r \in \mathcal{L}(R)$ or $\iota(w') \in N_{\mathfrak{I}}$. Then, by the inductive hypothesis, we immediately have that $\|R(w, w') \wedge A_1(w')\|_I \subseteq \mathsf{inst}_I$.
- $\top_r \notin \mathcal{L}(R)$ and $\iota(w') \notin N_{\mathfrak{I}}$. By the inductive hypothesis, a term w'' exists with $\iota(w'') = \iota(w')$ and $\|R(w, w'') \wedge A_1(w'')\|_I \subseteq \mathsf{inst}_I$.

In either cases, since $r \in \Xi_{\mathcal{K}}$ and I is closed under $\Xi_{\mathcal{K}}$, we have $\|A(w)\|_I \in \mathsf{inst}_I$.

Axioms of Type (6) and (9). Consider a rule r of the form $A_1(x) \rightarrow R(x, x) \wedge \mathbb{S}_R(x)$ in $\mathsf{D}_{\mathcal{K}}$ and consider two arbitrary terms w and w' occurring in I with $\iota(w) = \iota(w')$. Let σ be the substitution with $\sigma(x) = \iota(w)$, and assume that J_{n+1} is the result of applying $\langle r, \sigma \rangle$ to J_n. Then, we have that $\{A_1(\iota(w)), A_1(\iota(w))\} \subseteq \mathsf{inst}_{J_n}$, as well as $\mathsf{eq}_{J_{n+1}} = \mathsf{eq}_{J_n}$ and $\mathsf{inst}_{J_{n+1}} = \mathsf{inst}_{J_n} \cup \{\mathbb{S}_R(\iota(w)), R(\iota(w), \iota(w'))\}$. Since $\iota(w) = \iota(w')$, by the inductive hypothesis, we have that $\|A_1(w) \wedge A_1(w')\|_I \subseteq \mathsf{inst}_I$. Since $r \in \Xi_{\mathcal{K}}$ and I is closed under $\Xi_{\mathcal{K}}$, we have that $\|\mathbb{S}_R(w) \wedge \mathbb{S}_R(w')\|_I \subseteq \mathsf{inst}_I$ and $\|R(w, w) \wedge R(w', w')\|_I \subseteq \mathsf{inst}_I$; thus property (2) holds. We show that properties (3) and (4) hold by distinguishing three cases.

- $\top_r \in \mathcal{L}(R)$. As $\|A_1(w) \wedge A_1(w')\|_I \subseteq \mathsf{inst}_I$, $\Xi_{\mathcal{K}}$ contains rules (5.5) and (5.7), I is closed under $\mathsf{cl}_{\mathcal{K}}$, we have $\|\top_r(w, w')\|_I \in \mathsf{inst}_I$. As $\top_r \in \mathcal{L}(R)$ and I is closed under $\Xi_{\mathcal{K}}$, we have $\|R(w, w')\|_I \in \mathsf{inst}_I$.
- $\top_r \notin \mathcal{L}(R)$ and $\iota(w') \in N_{\mathfrak{I}}$. By the definition of ι and since $\iota(w) = \iota(w')$, we have that $w = w'$. Thus, we have $\|R(w, w')\|_I \in \mathsf{inst}_I$.
- $\top_r \notin \mathcal{L}(R)$ and $\iota(w') \notin N_{\mathfrak{I}}$. Then, by setting $w'' = w$, we have that $\iota(w'') = \iota(w')$ and $\|R(w, w'')\|_I \in \mathsf{inst}_I$.

Axioms of Type (8). Consider a rule $r \in \mathsf{D}_{\mathcal{K}}$ of the following form.

$$r = A(x) \wedge A(y) \wedge \mathbb{C}(x) \wedge \mathbb{C}(y) \wedge \bigwedge_{1 \leq i \leq k} \mathbb{C}(z_i) \wedge R_i(x, z_i) \wedge R_i(y, z_i) \rightarrow x \approx y$$

Furthermore, consider arbitrary terms $w_x, w_y, w_1, \ldots, w_k$ occurring in I and let σ be the following substitution.

$$\sigma(x) = \iota(w_x) \qquad \sigma(y) = \iota(w_y) \qquad \sigma(z_i) = \iota(w_i) \qquad 1 \leq i \leq k$$

Assume that J_{n+1} is the result of applying $\langle r, \sigma \rangle$ to J_n. Then, we have that $\sigma(\varphi) \subseteq \mathsf{inst}_{J_n}$ and $\sigma(x) \neq \sigma(y)$. Let w and w' be the $<$-larger and $<$-smaller terms between w_x and w_y; then J_{n+1} is obtained from J_n by replacing each occurrence of $\iota(w)$ in inst_{J_n} with $\iota(w')$, and by extending eq_{J_n} with $\iota(w) \rightsquigarrow \iota(w')$. Please note that for each variable z occurring in φ, we have that $\mathbb{C}(z) \in \varphi$; thus $\{w_x, w_y, w_1, \ldots, w_k\} \subseteq N_{\mathfrak{I}}$. Then, by the inductive hypothesis and because ι is the identity on $N_{\mathfrak{I}}$, we have that $\|\sigma(\varphi)\|_I \subseteq \mathsf{inst}_I$. Since $r \in \Xi_{\mathcal{K}}$ and I is closed under $\Xi_{\mathcal{K}}$, we have that $\|w_x\|_I = \|w_y\|_I$. By the definition of w and w', we also have that $\|w\|_I = \|w'\|_I$, and property (7) holds. Similarly as for the case of axioms of type (3), one can show that the properties are preserved by all the atoms in J_n that are replaced by the application of this rule.

Axioms of Type (10). Consider a rule r of the form $\bigwedge_{i=1}^{k} R_i(x_{i-1}, x_i) \rightarrow R(x_0, x_k)$ in $\mathsf{D}_{\mathcal{K}}$. Furthermore, for each $i \in [0..k]$, let w_i be an arbitrary term and let σ be the substitution with $\sigma(x_i) = w_i$. Assume that J_{n+1} is the result of applying $\langle r, \sigma \rangle$ to J_n. Then, for each $i \in [1..k]$, we have that $R_i(\iota(w_{i-1}), \iota(w_i)) \in \mathsf{inst}_{J_n}$; furthermore, we have $\mathsf{eq}_{J_{n+1}} = \mathsf{eq}_{J_n}$ and $\mathsf{inst}_{J_{n+1}} = \mathsf{inst}_{J_n} \cup \{R(\iota(w_0), \iota(w_k))\}$. Let $w_0'' = w_0$. By the inductive hypothesis, for each $i \in [1..k]$, a term w_i'' exists such that $\iota(w_i'') = \iota(w_i)$ and $\|R_i(w_{i-1}'', w_i'')\|_I \in \mathsf{inst}_I$. We distinguish three alternative cases.

- $\iota(w_k) \in N_{\mathfrak{I}}$. As $\iota(w_k) = \iota(w_k'')$, we have that $w_k = w_k''$. Since $r \in \Xi_{\mathcal{K}}$ and I is closed under $\Xi_{\mathcal{K}}$, we have $\|R(w_0, w_k)\|_I \in \mathsf{inst}_I$.

- $\top_r \in \mathcal{L}(R)$ and $\iota(w_k) \notin N_{\mathfrak{I}}$. Then, term w_k must have been generated by the application of a rule of the form $A_1(x) \rightarrow \exists z.S(x, z) \wedge \mathbb{D}_S(x, z) \wedge A(z)$ in $\Xi_{\mathcal{K}}$, and so

$\|A(w_k)\|_I \in \mathsf{inst}_I$. Since $\|R(w_0, w_1'')\|_I \in \mathsf{inst}_I$, $\Xi_\mathcal{K}$ contains rules (5.5)–(5.7), and I is closed under $\Xi_\mathcal{K}$, we have that $\|\top_r(w_0, w_k)\|_I \in \mathsf{inst}_I$. Since $\top_r \in \mathcal{L}(R)$ and I is closed under $\Xi_\mathcal{K}$, we also have $\|R(w_0, w_k)\|_I \in \mathsf{inst}_I$

- $\top_r \notin \mathcal{L}(R)$ and $\iota(w_k) \notin N_\mathfrak{I}$. Because r occurs in $\Xi_\mathcal{K}$ and I is closed under $\Xi_\mathcal{K}$, we have that $\|R(w_0, w_k'')\|_I \in \mathsf{inst}_I$ and property (4) holds.

Rules of the form (5.2). Consider a rule r of the form $\mathbb{C}(x) \wedge R(x, x) \to \mathbb{S}_R(x)$ in $\mathsf{D}_\mathcal{K}$, and consider an arbitrary term w occurring in I. Let σ be the substitution with $\sigma(x) = \iota(w)$, and assume that J_{n+1} is the result of applying $\langle r, \sigma\rangle$ to J_n. Then, we have that $\{\mathbb{C}(\iota(w)), R(\iota(w), \iota(w))\} \subseteq \mathsf{inst}_{J_n}$. Since $\iota(w)$ satisfies $\mathbb{C}$, we have $\iota(w) \in N_\mathfrak{I}$ and, by the inductive hypothesis, we have $\|\mathbb{C}(w) \wedge R(w, w)\|_I \subseteq \mathsf{inst}_I$. As $r \in \Xi_\mathcal{K}$ and I is closed under $\Xi_\mathcal{K}$, we have $\|\mathbb{S}_R(w)\|_I \in \mathsf{inst}_I$.

Rules of the form (5.3). Consider a rule r of the form $\mathbb{S}_S(x) \to \mathbb{S}_R(x)$ in $\mathsf{D}_\mathcal{K}$, and consider an arbitrary term w occurring in I. Let σ be the substitution with $\sigma(x) = \iota(w)$, and assume that J_{n+1} is the result of applying $\langle r, \sigma\rangle$ to J_n. Then, we have $\mathbb{S}_S(\iota(w)) \in \mathsf{inst}_{J_n}$. By the inductive hypothesis, we have $\|\mathbb{S}_S(w) \wedge S(w, w)\|_I \subseteq \mathsf{inst}_I$. By the definition of $\mathsf{cl}_\mathcal{K}$, rule base $\Xi_\mathcal{K}$ contains a rule r' of the form $S(x, y) \to R(x, y)$. Since $\Xi_\mathcal{K}$ contains also rule r and I is closed under $\Xi_\mathcal{K}$, we have that $\|\mathbb{S}_R(w) \wedge R(w, w)\|_I \subseteq \mathsf{inst}_I$.

Rules of the form (5.4). Consider a rule r of the form $\mathbb{D}_S(x, y) \to \mathbb{D}_R(x, y)$ in $\mathsf{D}_\mathcal{K}$, and consider an arbitrary term w occurring in I and an arbitrary auxiliary constant $o_{P,B}$. Let σ be the substitution with $\sigma(x) = \iota(w)$ and $\sigma(y) = o_{P,B}$, and assume that J_{n+1} is the result of applying $\langle r, \sigma\rangle$ to J_n. Then, we have that $\mathbb{D}_S(\iota(w), o_{P,B}) \in \mathsf{inst}_{J_n}$. By the inductive hypothesis, we have that $P \sqsubseteq^*_\mathcal{R} S$, and a term w^* exists such that $\iota(w^*) = o_{P,B}$ and $\|P(w, w^*) \wedge \mathbb{D}_P(w, w^*)\|_I \subseteq \mathsf{inst}_I$. By the definition of $\mathsf{cl}_\mathcal{K}$, we have that $S \sqsubseteq R \in \mathcal{R}$; thus $P \sqsubseteq^*_\mathcal{R} R$.

Rules of the form (5.6). Consider a rule r of the form $R(x, y) \to \top_c(x) \wedge \top_c(y)$ in $\mathsf{D}_\mathcal{K}$, and consider arbitrary terms w and w' occurring in I. Let σ be the substitution with $\sigma(x) = \iota(w)$ and $\sigma(y) = \iota(w')$, and assume that J_{n+1} is the result of applying $\langle r, \sigma\rangle$ to J_n. Then, we have that $R(\iota(w), \iota(w')) \in \mathsf{inst}_{J_n}$, $\mathsf{eq}_{J_{n+1}} = \mathsf{eq}_{J_n}$, and $\mathsf{inst}_{J_{n+1}}$ extends inst_{J_n} with

$\top_c(\iota(w))$ and $\top_c(\iota(w'))$. We distinguish two cases.

- $\top_r \in \mathcal{L}(R)$ or $\iota(w') \in N_\mathfrak{I}$. By the inductive hypothesis, we have $\|R(w, w')\|_I \in \mathsf{inst}_I$. Since $r \in \Xi_\mathcal{K}$ and I is closed under I, we have that $\|\top_c(w) \wedge \top_c(w')\|_I \subseteq \mathsf{inst}_I$.

- $\top_r \notin \mathcal{L}(R)$ and $\iota(w') \notin N_\mathfrak{I}$. Hence, $\iota(w')$ is of the form $o_{P,B}$. By the inductive hypothesis, a term w'' exists such that $\iota(w'') = o_{P,B}$ and $\|R(w, w'')\|_I \in I$. Since $\Xi_\mathcal{K}$ contains rule r and I is closed under $\Xi_\mathcal{K}$, we have that $\|\top_c(w)\|_I \in \mathsf{inst}_I$. We are left to show that the same holds for w'. To this end, please note that term w' must have been generated by the application of a rule in $\Xi_\mathcal{K}$ of the form $A'_1(x) \rightarrow \exists z.P(x, z) \wedge \mathbb{D}_P(x, z) \wedge B(z)$; thus $\|B(w')\|_I \in \mathsf{inst}_I$; furthermore, $\Xi_\mathcal{K}$ contains the rule $B(x) \rightarrow \top_c(x)$. Since I is closed under $\Xi_\mathcal{K}$, we have $\|\top_c(w')\|_I \in \mathsf{inst}_I$.

Rule of the form (5.7). Consider the rule r of the form $\top_c(x) \wedge \top_c(y) \rightarrow \top_r(x, y)$ in $\mathsf{D}_\mathcal{K}$, and consider arbitrary terms w and w' occurring in I. Let σ be the substitution with $\sigma(x) = \iota(w)$ and $\sigma(y) = \iota(w')$, and assume that J_{n+1} is the result of applying $\langle r, \sigma \rangle$ to J_n. Then, we have $\{\top_c(\iota(w)), \top_c(\iota(w'))\} \subseteq \mathsf{inst}_{J_n}$, $\mathsf{eq}_{J_{n+1}} = \mathsf{eq}_{J_n}$, and $\mathsf{inst}_{J_{n+1}}$ extends inst_{J_n} with $\top_r(\iota(w), \iota(w'))$. By the inductive hypothesis, we have $\|\top_c(w) \wedge \top_c(w')\|_I \subseteq \mathsf{inst}_I$. Since $r \in \Xi_\mathcal{K}$ and I is closed under $\Xi_\mathcal{K}$, we have that $\|\top_r(w, w')\|_I \in \mathsf{inst}_I$.

Rule of the form (5.8). Consider a rule r of the form $\bot_r(x, y) \rightarrow \bot_c(x)$ in $\mathsf{D}_\mathcal{K}$, and consider arbitrary terms w and w' occurring in I. Let σ be the substitution with $\sigma(x) = \iota(w)$ and $\sigma(y) = \iota(w')$, and assume that J_{n+1} is the result of applying $\langle r, \sigma \rangle$ to J_n. Then, we have that $\bot_r(\iota(w), \iota(w')) \in \mathsf{inst}_{J_n}$, $\mathsf{eq}_{J_{n+1}} = \mathsf{eq}_{J_n}$, and $\mathsf{inst}_{J_{n+1}}$ extends inst_{J_n} with $\bot_c(\iota(w))$. We distinguish two cases.

- $\top_r \in \mathcal{L}(\bot_r)$ or $\iota(w') \in N_\mathfrak{I}$. By the inductive hypothesis, we have $\|\bot_r(w, w')\|_I \in \mathsf{inst}_I$.

- $\top_r \notin \mathcal{L}(\bot_r)$ and $\iota(w') \notin N_\mathfrak{I}$. By the inductive hypothesis, a term w'' exists such that $\iota(w'') = \iota(w')$ and $\|R(w, w'')\|_I \in I$.

Since $r \in \Xi_\mathcal{K}$ and I is closed under $\Xi_\mathcal{K}$, in either cases we have that $\|\bot_c(w)\|_I \in \mathsf{inst}_I$. □

Equality rules occurring in $\Xi_\mathcal{K}$ and $\mathsf{D}_\mathcal{K}$ can only replace terms with individuals from $N_\mathfrak{I}$; more formally, for each $w \rightsquigarrow w' \in I \cup J$, we have $w' \in N_\mathfrak{I}$. Thus, together property (3) of

Lemma 5.9 and property (7) of Lemma 5.10 imply the following result.

Lemma 5.11. *For each term w occurring in I and each individual $a \in N_{\mathfrak{I}}$,*

(e1) $\|w\|_I = a$ *if and only if* $\|\iota(w)\|_J = a$, *and*

(e2) $\|w\|_I = w$ *if and only if* $\|\iota(w)\|_J = \iota(w)$.

This result together with properties (1) and (2) of Lemma 5.9 imply that ι is an homomorphism from inst_I to inst_J.

Lemma 5.12. *Function ι satisfies the following three properties for all terms w and w', each $A \in \{\mathbb{C}\} \cup \mathsf{con}_{\mathcal{K}} \cup \{\mathbb{S}_S \mid S \in \mathsf{rol}_{\mathcal{K}}\}$, and each $R \in \mathsf{rol}_{\mathcal{K}} \cup \{\mathbb{D}_S \mid S \in \mathsf{rol}_{\mathcal{K}}\}$.*

(h1) $A(w) \in \mathsf{inst}_I$ *implies that* $A(\iota(w)) \in \mathsf{inst}_J$.

(h2) $R(w, w') \in \mathsf{inst}_I$ *implies that* $R(\iota(w), \iota(w')) \in \mathsf{inst}_J$.

In addition, Lemma 5.11 together with properties (1)–(6) of Lemma 5.10 imply that ι establishes a tight connection from inst_J to inst_I, as shown by Lemma 5.13. Note that, as I is closed under $\Xi_{\mathcal{K}}$, for all roles P and S, and all term w and w', we have that $\{P(w, w'), \mathbb{D}_P(w, w')\} \subseteq \mathsf{inst}_I$ and $P \sqsubseteq^*_{\mathcal{R}} S$ imply that $\{S(w, w'), \mathbb{D}_S(w, w')\} \subseteq \mathsf{inst}_I$. Thus, property (d5) follows from property (5) of Lemma 5.10 and Lemma 5.11.

Lemma 5.13. *Function ι satisfies the following properties for all terms w and w' occurring in I, each constant $o_{P,B} \in N_{aux}$, each concept $A \in \{\mathbb{C}\} \cup \mathsf{con}_{\mathcal{K}}$, and each role $R \in \mathsf{rol}_{\mathcal{K}}$.*

(d1) $A(\iota(w)) \in \mathsf{inst}_J$ *implies that* $A(w) \in \mathsf{inst}_I$.

(d2) $\mathbb{S}_R(\iota(w)) \in \mathsf{inst}_J$ *implies that* $\mathbb{S}_R(w) \wedge R(w, w) \in \mathsf{inst}_I$.

(d3) $R(\iota(w), \iota(w')) \in \mathsf{inst}_J$, *and* $\top_r \in \mathcal{L}(R)$ *or* $\iota(w') \in N_{\mathfrak{I}}$ *imply that* $R(w, w') \in \mathsf{inst}_I$.

(d4) $R(\iota(w), \iota(w')) \in \mathsf{inst}_J$, $\top_r \notin \mathcal{L}(R)$, *and* $\iota(w') \notin N_{\mathfrak{I}}$ *imply that a term w'' occurring in I exists such that* $\iota(w'') = \iota(w')$ *and* $R(w, w'') \in \mathsf{inst}_I$.

(d5) $\iota(w)$ *is directly connected to* $o_{P,B}$ *in J implies that a term w^* exists in I such that*

- $\iota(w^*) = o_{P,B}$, *and*

- $\{S(w, w^*), \mathbb{D}_S(w, w^*)\} \subseteq \mathsf{inst}_I$ *for each role* $S \in \mathsf{rol}_{\mathcal{K}}$ *with* $\mathbb{D}_S(\iota(w), o_{P,B}) \in \mathsf{inst}_J$.

(d6) For each constant u *occurring in* J*, a term* w_u *exists such that* $\iota(w_u) = u$.

Finally, property (a) follows immediately from Lemmas 5.11, 5.12, and 5.13.

Chapter 6

Answering CQs over $\mathcal{ELHO}_\bot^+$ KBs

While the datalog program $\mathsf{D}_\mathcal{K}$ for an $\mathcal{ELHO}_\bot^+$ knowledge base $\mathcal{K}$ can be directly used to answer instance queries, answering a conjunctive query q over $\mathsf{D}_\mathcal{K}$ can produce unsound answers. In this chapter, we present an NP filtering procedure that checks whether a candidate answer τ to q over $\mathsf{D}_\mathcal{K}$ is sound. Towards this goal, in Section 6.1 we discuss the intuitions behind our filtering procedure; and in Section 6.2 we introduce the procedure formally and show that it runs in nondeterministic polynomial time in the combined size of $\mathcal{K}$, q, and τ. We also prove that the procedure runs in nondeterministic polynomial time only for candidate answers that map query variables to auxiliary constants, and depend on transitive or reflexive roles—in particular, filtering can be done in polynomial time if $\mathcal{K}$ is in $\mathcal{ELHO}_\bot$. Then, we show how to use our filtering procedure to obtain the first worst-case optimal algorithm for answering conjunctive queries over $\mathcal{ELHO}_\bot^+$ knowledge bases that runs in NP in combined complexity and in polynomial time in knowledge base complexity, thus we settle the open questions of the complexity of CQ answering for $\mathcal{EL}$ variants with transitive roles. Finally, in Section 6.3 we prove that our filtering procedure is worst-case optimal, whereas in Section 6.4 we prove the procedure's correctness.

6.1 Intuition

Algorithm 1 on page 93 specifies a procedure $\mathsf{isSound}_{TR}(q, \mathsf{D}_\mathcal{K}, \tau)$ that checks whether candidate answer τ to q over $\mathsf{D}_\mathcal{K}$ is sound. In step 1, our procedure uses an auxiliary procedure

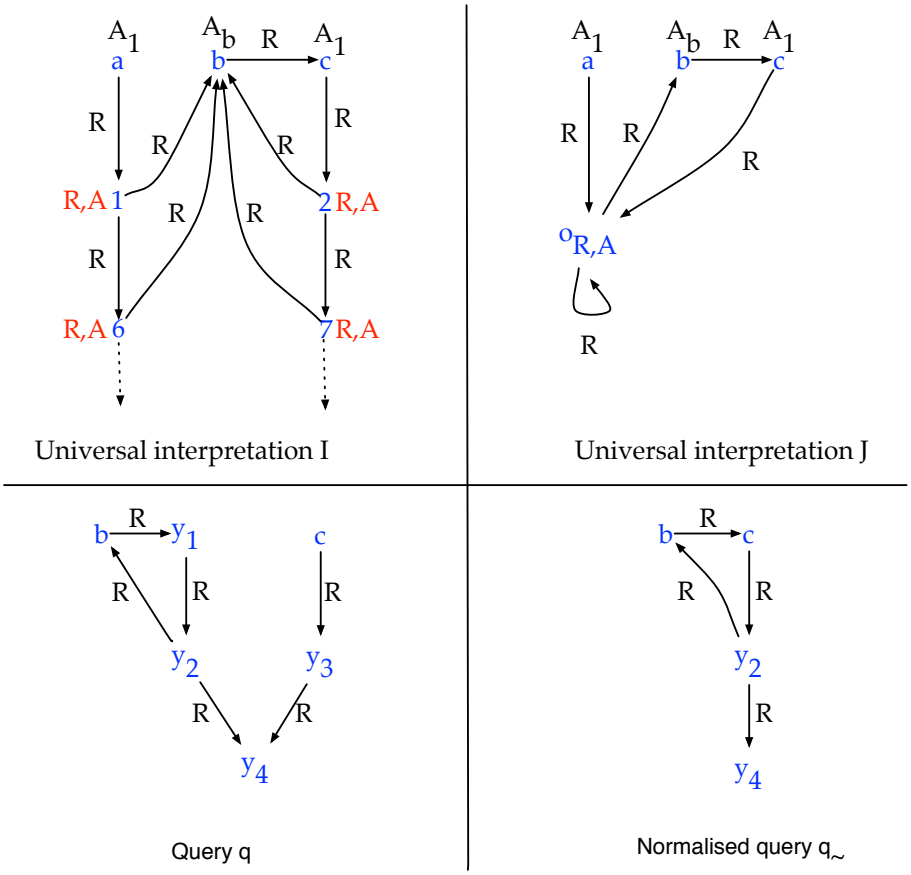

Figure 6.1: The universal interpretations I and J of $\Xi_{\mathcal{K}}$ and $\mathsf{D}_{\mathcal{K}}$ for the KB $\mathcal{K}$ from Example 4.3, and the query q and the normalised query $q_{\sim}$

$\mathsf{isDSound}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ that checks in polynomial time necessary conditions for the soundness of τ. When τ does not map the variables of q to auxiliary constants, or does not depend on transitive and reflexive roles, step 1 suffices to determine whether τ is sound. For all other cases, the procedure uses additional steps to determine the soundness of τ.

6.1.1 Filtering without Transitive and Reflexive Roles

We first discuss the intuitions behind the auxiliary procedure $\mathsf{isDSound}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ using the $\mathcal{ELHO}_{\bot}$ knowledge base from Example 4.3 on page 42, and the query q and the candidate answer τ from Example 6.1.

Example 6.1. *Let $\mathcal{K}$ be the $\mathcal{ELHO}_{\bot}$ KB from Example 4.3 and let q and τ be the following*

Boolean CQ and substitution, respectively.

$$q = \exists \vec{y}.\ R(b, y_1) \wedge R(y_1, y_2) \wedge R(y_2, b) \wedge R(c, y_3) \wedge R(y_2, y_4) \wedge R(y_3, y_4)$$

$$\tau = \{y_1 \mapsto c,\ y_2 \mapsto o_{R,A},\ y_3 \mapsto o_{R,A},\ y_4 \mapsto o_{R,A}\}$$

The upper part of Figure 6.1 shows the universal interpretations I and J of $\Xi_{\mathcal{K}}$ and $\mathsf{D}_{\mathcal{K}}$, respectively. Please note that both interpretations contain only direct edges. Moreover, the lower left part of Figure 6.1 shows the query q. One can easily check that $\mathsf{D}_{\mathcal{K}} \models \tau(q)$.

We next show how $\mathsf{isSound}_{TR}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ decides that τ is sound—that is, that a substitution π mapping the variables in q to elements in I exists such that $\pi(q) \subseteq I$. The knowledge base $\mathcal{K}$ does not contain transitive and reflexive roles, so the soundness of τ can be determined using solely the auxiliary procedure $\mathsf{isDSound}(q, \mathsf{D}_{\mathcal{K}}, \tau)$.

We use substitution τ to restrict the possible form of π. Variable y_1 must be mapped by π to individual c, since $\tau(y_1) = c$; in contrast, substitution π must map variables y_2, y_3, and y_4 to an arbitrary labelled null of type R, A occurring in I, since $\tau(y_2) = \tau(y_3) = \tau(y_4) = o_{R,A}$.

Each such substitution π is guaranteed to satisfy all unary atoms of q, all binary atoms of q that contain a role that is implied by the top role, all binary atoms of q that τ maps to edges pointing towards (non-auxiliary) individuals from $N_{\mathfrak{I}}$, and all binary atoms of q that contain a single variable and that τ maps to self edges in J. We call each atom of q that satisfies at least one of these conditions *good*. In our example, atoms $R(b, y_1)$ and $R(y_2, b)$ are good because τ maps them onto edges pointing towards individuals from $N_{\mathfrak{I}}$; hence, to show that τ is sound, we must demonstrate that a substitution π exists that satisfies the non-good atoms of q in I.

Function $\mathsf{isDSound}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ implements a revised version of the 'fork' and 'acyclicity' checks by Lutz et al. [72] described in Section 4.1. To guarantee completeness, we consider only those binary atoms in q that contain simple roles, and thus cannot be expanded into role chains using complex role inclusions, and that τ maps onto direct edges in J pointing towards auxiliary constants. We call these atoms *aux-simple* as they can be mapped onto the direct edges in I pointing towards labelled nulls; apart from atoms $R(b, y_1)$ and $R(y_2, b)$ which are good, all other atoms of q are aux-simple.

The checks in step 1 are based on computing equality constraints of the form $s \sim t$ over the terms in the query. Such a constraint asserts that τ must map terms s and t to equal constants in $\mathsf{D}_{\mathcal{K}}$—that is, $\mathsf{D}_{\mathcal{K}} \models \tau(s) \approx \tau(t)$. Example 6.2 explains how to derive the equality constraints for q and τ using the 'aux-simple forks' in the query.

Example 6.2. *The equality constraints for q and τ are as follows.*

$$y_2 \sim y_3 \tag{6.1}$$

$$y_1 \sim c \tag{6.2}$$

Atoms $R(y_2, y_4) \wedge R(y_3, y_4)$ in q form a 'fork' that involves only aux-simple atoms. Moreover, substitution τ maps variable y_4 to auxiliary constant $o_{R,A}$; this constant represents the labelled nulls of type R, A in I. Because these atoms are aux-simple, each substitution π must map these atoms onto direct edges in I pointing towards labelled nulls. However, the direct edges that point towards labelled nulls in I induce a forest-shaped structure, so y_2 and y_3 must be mapped to the same element (in both I and J). This is captured by the equality constraint (6.1). Equality constraints, moreover, need to be propagated further down in the query. In our example, due to (6.1), atoms $R(y_1, y_2) \wedge R(c, y_3)$ also constitute a 'fork' since τ maps y_2 and y_3 to auxiliary constants; so we derive equality constraint (6.2) as well.

Then, in step 1 we compute a new query $q_{\sim}$ by 'applying' the equality constraints to q; Figure 6.1 shows the resulting query $q_{\sim}$ in which variables y_1 and y_3 are replaced with individual c and variable y_2, respectively. Finally, function $\mathsf{isDSound}$ checks that (i) substitution τ satisfies the derived equality constraints, and (ii) query $q_{\sim}$ is 'aux-acyclic'—that is, $q_{\sim}$ does not contain cycles consisting only of aux-simple atoms. In our example, substitution τ clearly satisfies the equality constraints (6.1) and (6.2) since $\tau(y_2) = \tau(y_3)$ and $\tau(y_1) = c$. Moreover, atoms $R(b, c) \wedge R(c, y_2) \wedge R(y_2, b)$ form a cycle in $q_{\sim}$, but this cycle involves good atoms; hence, function $\mathsf{isDSound}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ returns t. Because q contains only good and aux-simple atoms, substitution τ is sound. For example, using Figure 6.1, one can check that the following substitution embeds q into I.

$$\pi = \{y_1 \mapsto c,\ y_2 \mapsto 2,\ y_3 \mapsto 2,\ y_4 \mapsto 7\}$$

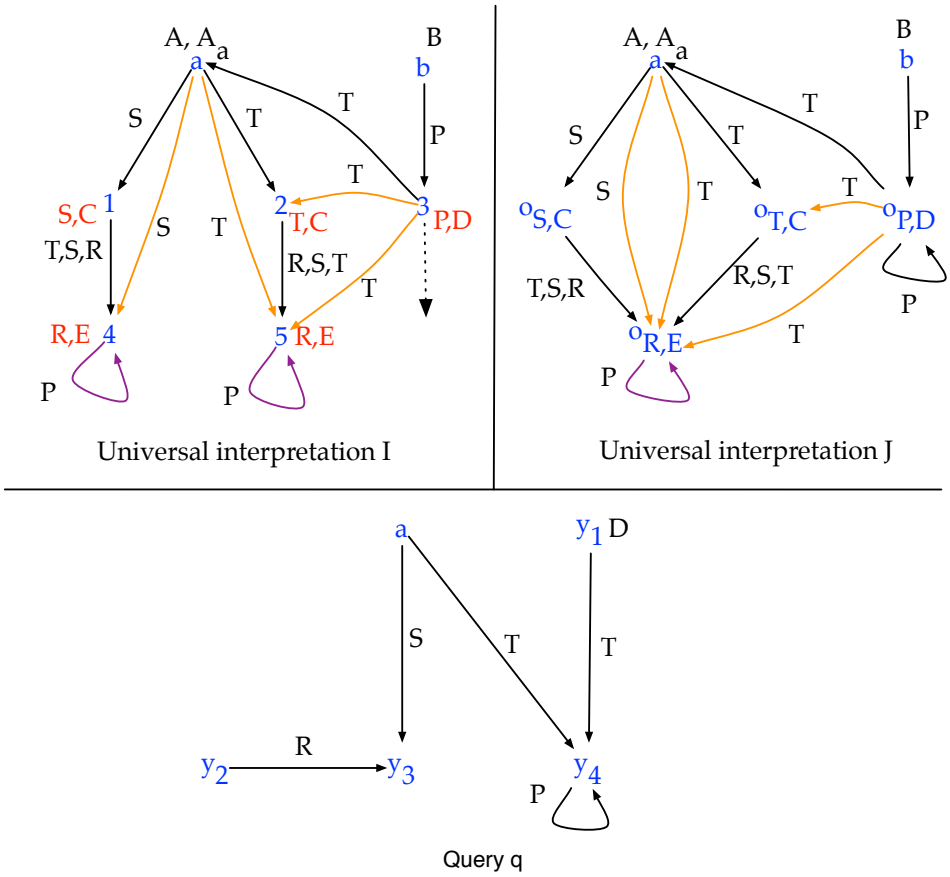

Figure 6.2: The universal interpretations I and J of $\Xi_{\mathcal{K}}$ and $\mathsf{D}_{\mathcal{K}}$ for the knowledge base from Example 4.1, and the query q

6.1.2 Filtering with Transitive and Reflexive Roles

The auxiliary procedure $\mathsf{isDSound}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ cannot determine the soundness of τ when q contains atoms that are neither good nor aux-simple. In this case, the filtering procedure $\mathsf{isSound}_{TR}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ uses steps 3–21 to ensure that a substitution π exists that embeds all atoms of q into the universal interpretation of $\mathcal{K}$. We next discuss the intuitions behind these additional steps using the $\mathcal{ELHO}_{\bot}^{+}$ knowledge base from Example 4.1 on page 39, and the query q and the candidate answer τ from Example 6.3.

Example 6.3. *Let $\mathcal{K}$ be the $\mathcal{ELHO}_{\bot}^{+}$ KB from Example 4.1, and let q and τ be the following*

Boolean CQ and substitution, respectively.

$$q = \exists \vec{y}.\ S(a, y_3) \wedge R(y_2, y_3) \wedge T(a, y_4) \wedge P(y_4, y_4) \wedge T(y_1, y_4) \wedge D(y_1)$$

$$\tau = \{y_1 \mapsto o_{P,D},\ y_2 \mapsto o_{S,C},\ y_3 \mapsto o_{R,E},\ y_4 \mapsto o_{R,E}\}$$

The upper part of Figure 6.2 shows the universal interpretations I and J of $\Xi_{\mathcal{K}}$ and $\mathsf{D}_{\mathcal{K}}$, respectively. The lower part of Figure 6.2 shows the CQ q; one can check that $\mathsf{D}_{\mathcal{K}} \models \tau(q)$.

We next show how $\mathsf{isSound}_{TR}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ decides that τ is sound—that is, that a substitution π mapping the variables in q to the elements in I exists such that $\pi(q) \subseteq I$. We again use τ to restrict the search space for such a substitution π. In particular, π must map variable y_1 to term 3 or one of its successors in I because $\tau(y_1) = o_{P,D}$, variable y_2 to term 1 because $\tau(y_2) = o_{S,C}$, and variables y_3 and y_4 to one of 4 and 5 because $\tau(y_3) = \tau(y_4) = o_{R,E}$.

The good atoms of q will be satisfied by any such substitution π. In our example, atom $D(y_1)$ is good because it is unary, whereas atom $P(y_4, y_4)$ is good because it contains a single variable and it is mapped to the self edge in J. We are left to show that a substitution π exists that satisfies all of the other atoms of q.

Step 1 of Algorithm 1 checks that the aux-simple atoms of q can be mapped onto the direct edges pointing towards labelled nulls occurring in I. We first use the 'aux-simple forks' in q to compute equality constraints; then, we check that these constraints are satisfied by q and τ, we apply the constraints to obtain a new query $q_{\sim}$, and we check that $q_{\sim}$ does not contain cycles consisting only of aux-simple atoms. The query in our example contains a single aux-simple atom $R(y_2, y_3)$, so $q_{\sim} = q$ and the auxiliary procedure $\mathsf{isDSound}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ returns t. Atoms $S(a, y_3)$, $T(a, y_4)$, and $T(y_1, y_4)$ are neither good nor aux-simple, so we proceed to step 3.

A substitution π that embeds $q_{\sim}$ into I can map distinct variables of $q_{\sim}$ to the same labelled null of I, if τ also maps these variables to the same auxiliary constant in J. In step 3, we take this into account and guess a renaming σ for the variables in $q_{\sim}$; so in the rest of Algorithm 1 we consider $\sigma(q_{\sim})$ instead of $q_{\sim}$. In our example, we guess σ to be identity,

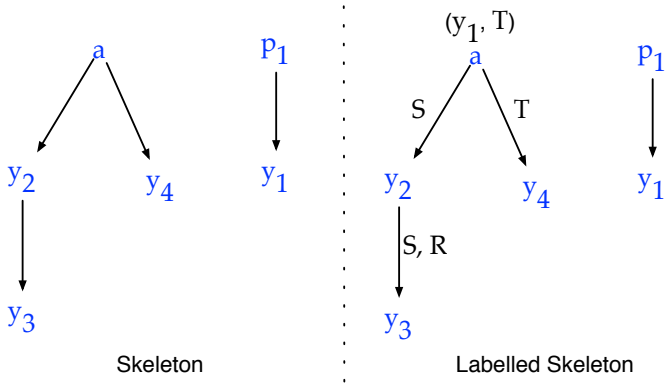

Figure 6.3: The skeleton $\mathcal{S}$ for $\sigma(q_\sim)$

so $\sigma(q_\sim) = q_\sim$.

In step 4, we guess a *skeleton* $\mathcal{S}$ for $\sigma(q_\sim)$, which is a finite structure that finitely describes the (possibly infinite) set of all substitutions π mapping the variables in $\sigma(q_\sim)$ to distinct labelled nulls of I. The vertices of $\mathcal{S}$ are the individuals occurring in $\sigma(q_\sim)$, the variables of $\sigma(q_\sim)$ that τ maps to auxiliary constants, and a special *placeholder variable* p_i for each variable y_i occurring in $\sigma(q_\sim)$. We use the placeholder variables to compactly represent individuals occurring in I that do not occur in the query. The skeleton's vertices are arranged in a forest whose roots are the individuals occurring in $\sigma(q_\sim)$ and the placeholder variables. Figure 6.3 shows the skeleton $\mathcal{S}$ for our example query; the roots of $\mathcal{S}$ that do not have any descendants are not shown for clarity. Such $\mathcal{S}$ represents those substitutions π that map

(1) each placeholder variable p_i to some named individual from $N_\mathfrak{I}$;

(2) variable y_1 to a labelled null of I under named individual $\pi(p_1)$;

(3) variables y_2 and y_4 to labelled nulls under individual a; and,

(4) variable y_3 to a labelled null of I under labelled null $\pi(y_2)$.

The constraints expressed by the skeleton $\mathcal{S}$, however, do not sufficiently describe the substitutions that map onto I the binary atoms of $\sigma(q_\sim)$ that are neither good nor aux-simple.

For example, skeleton $\mathcal{S}$ represents substitution

$$\pi_1 = \{p_1 \mapsto b,\ y_1 \mapsto 3,\ y_2 \mapsto 1,\ y_3 \mapsto 4,\ y_4 \mapsto 4\}$$

but $T(\pi_1(y_1), \pi_1(y_4)) \notin I$.

To eliminate such substitutions, we refine the skeleton's constraints in steps 5–15 of Algorithm 1. We guess for each binary atom in $\sigma(q)$ that is neither good nor aux-simple how to unfold it as a sequence of direct edges in I using the transitive roles in $\mathcal{K}$. The direct paths in I can be of two types: *anonymous paths* consist only of direct edges pointing to labelled nulls, whereas *nominal paths* contain at least one direct edge pointing to an individual. For example, edge $S(a, 4)$ can be unfolded into the anonymous path $\rho_S = S \cdot S$ connecting a with 1 and 1 with 4. In contrast, edge $T(3, 5)$ can be unfolded into the nominal path $\rho_T = T \cdot T \cdot T$ connecting 3 with individual a, a with 2, and 2 with 5.

The skeleton already constrains the relative positions of query terms; so we unfold binary atoms in accordance with these positions. In particular, binary atoms whose terms are reachable in $\mathcal{S}$ will be unfolded using an anonymous path, whereas binary atoms whose terms are not reachable in $\mathcal{S}$ will be unfolded using a nominal path. We represent the unfolding of each binary atom by labelling each edge $\langle v, v' \rangle \in \mathcal{S}$ with a set of roles $L(v, v')$, and each root v_r of $\mathcal{S}$ with a set $L(v_r)$ of pairs $\langle s, P \rangle$ consisting of a term s and of a role P; after these steps, $\mathcal{S}$ represents those substitutions π that satisfy the following two properties.

(5) For each role $P \in L(v, v')$, a non-empty role chain $\rho \in \mathcal{L}(P)$ exists labelling an anonymous path in I from $\pi(v)$ to $\pi(v')$.

(6) For each pair $\langle s, P \rangle \in L(v_r)$, an edge from $\pi(s)$ to $\pi(v_r)$ in I exists that is labelled by role P.

We next illustrate how to label the skeleton so that all substitutions represented by $\mathcal{S}$ satisfy the binary atoms of $\sigma(q_\sim)$.

In step 6, the aux-simple atoms in $\sigma(q_\sim)$ are used to label the skeleton. The query in our example contains a single aux-simple atom $R(y_2, y_3)$, so we label the skeleton edge $\langle y_2, y_3 \rangle$ with role R. Next, we proceed to step 7–15 and show how the other binary atoms of $\sigma(q_\sim)$

contribute to the labelling of $\mathcal{S}$.

For atom $S(a, y_3)$, we must ensure that, for each substitution π represented by $\mathcal{S}$, a role chain $\rho_S \in \mathcal{L}(S)$ exists connecting a to $\pi(y_3)$ using only direct edges. Now, individual a reaches variable y_3 in $\mathcal{S}$ via variable y_2, so ρ_S must connect a to $\pi(y_2)$, and $\pi(y_2)$ to $\pi(y_3)$ without going through individuals. Hence, in step 8 we guess a transitive role P such that $P \sqsubseteq^*_{\mathcal{R}} S$ and $\rho_S \in \mathcal{L}(P)$. In steps 10–11 we then 'split' atom $P(a, y_3)$ into atoms $P(a, y_2)$ and $P(y_2, y_3)$: atom $P(a, y_2)$ captures the anonymous subpath of ρ_S connecting a with $\pi(y_2)$, whereas atom $P(y_2, y_3)$ captures the anonymous subpath of ρ_S connecting $\pi(y_2)$ with $\pi(y_3)$. We do not know to which elements of I we should map variables y_2 and y_3 to, so we cannot check the existence of the required subpaths independently. Therefore, we add in step 15 role P as constraints on edges $\langle a, y_2\rangle$ and $\langle y_2, y_3\rangle$ in $\mathcal{S}$. Each skeleton edge $\langle v, v'\rangle$ thus 'accumulates' all constraints that the anonymous path connecting $\pi(v)$ to $\pi(v')$ must satisfy. Later we shall explain how in steps 18–27 we check these constraints and, if this check passes, how we know that we can map y_2 and y_3 to labelled nulls below individual a. Because S is a transitive role, in our example we can guess $P = S$.

For atom $T(y_1, y_4)$, we must ensure that, for each substitution π represented by $\mathcal{S}$, a role chain $\rho_T \in \mathcal{L}(T)$ exists connecting $\pi(y_1)$ to $\pi(y_4)$ using only direct edges. Since the relative positions of $\pi(y_1)$ and $\pi(y_4)$ in I are determined by $\mathcal{S}$ as shown in Figure 6.3, such a path must connect $\pi(y_1)$ with a and then connect a with $\pi(y_4)$. In addition, we can assume that the subpath from a to $\pi(y_4)$ is anonymous: if a path from $\pi(y_1)$ to $\pi(y_4)$ involves individuals other than a or if it visits a more than once, we can 'absorb' all such path segments into the subpath from $\pi(y_1)$ to a. Hence, in step 8 we guess a transitive role P such that $P \sqsubseteq^*_{\mathcal{R}} T$ and $\rho_T \in \mathcal{L}(P)$. In steps 10–11 we 'split' atom $P(y_1, y_4)$ into atoms $P(y_1, a)$ and $P(a, y_4)$: atom $P(y_1, a)$ captures the nominal subpath of ρ_T connecting $\pi(y_1)$ with a, whereas atom $P(a, y_4)$ captures the anonymous subpath of ρ_T connecting a with $\pi(y_4)$. In step 14, we add pair $\langle y_1, P\rangle$ as a constraint on the individual a; thus the roots of $\mathcal{S}$ 'accumulate' all constraints that the nominal paths from labelled nulls to individuals must satisfy. In step 15, we instead add P as a constraint on edge $\langle a, y_4\rangle$. Because T is a transitive role, in our example we can again guess $P = T$. Finally, as a is not connected to $\pi_1(y_4)$ via a sequence

of T edges, adding T as constraint on edge $\langle a, y_4 \rangle$ ensures that substitution π_1 does not satisfy property (5).

Finally, for atom $T(a, y_4)$, we must ensure that, for each substitution π represented by $\mathcal{S}$, a role chain $\rho_T \in \mathcal{L}(T)$ exists connecting a to $\pi(y_4)$ using only direct edges. Since a directly reaches y_4 in $\mathcal{S}$, such a path must connect a with $\pi(y_4)$ without going through individuals. Hence, in step 8 we guess a transitive role P such that $P \sqsubseteq^*_{\mathcal{R}} T$ and $\rho_T \in \mathcal{L}(P)$. In steps 10–11, no splitting is necessary because a and y_4 are directly connected in $\mathcal{S}$—that is, we use atom $P(a, y_4)$ to represent the anonymous path ρ_T connecting a with $\pi(y_4)$. In step 15, we then add P as a constraint on edge $\langle a, y_4 \rangle$. Because T is a transitive role, in our example we can guess $P = T$.

After the for-loop in steps 7–15, the labelled skeleton $\mathcal{S}$ shown in Figure 6.3 represents all substitutions satisfying properties (5) and (6). Although each such substitution maps $\sigma(q_\sim)$ onto I, we must still show that at least one such substitution π can be realised by the universal interpretation I. However, a terminating algorithm cannot materialise the universal interpretation I, so we exploit a property of our datalog program $\mathsf{D}_\mathcal{K}$: an element w is connected to an element w' in I using a direct edge labelled by R if and only if $\mathsf{D}_\mathcal{K} \models \mathbb{D}_R(u, u')$ where u and u' are the datalog constants representing elements w and w' of I, respectively. Therefore, a substitution π satisfies properties (5) and (6) if and only if the following two properties hold for each edge $\langle v, v' \rangle$ and each root vertex v_r of $\mathcal{S}$.

(a) For each role $P \in L(v, v')$, a non-empty role chain $\rho \in \mathcal{L}(P)$ exists labelling a path from u to u' in J consisting only of direct edges pointing to auxiliary constants, where u and u' are the constants representing $\pi(v)$ and $\pi(v')$ in J, respectively.

(b) For each $\langle s, P \rangle \in L(v_r)$, there exists an edge labelled by P from u_s to u_r in J, where u_s and u_r are the constants representing $\pi(s)$ and $\pi(v_r)$ in J, respectively.

These properties provide us with an effective way of checking the skeleton's constraints by applying the procedure check_{TR} from Definition 6.9 to each skeleton vertex $v \in \mathcal{V}$. We check property (b) directly in function check_{TR} by testing entailments over $\mathsf{D}_\mathcal{K}$. In contrast, we check property (a) by applying the auxiliary function exist_{TR} from Definition 6.8 to each

edge $\langle v, v' \rangle$ in $\mathcal{S}$. Using the direct predicates in $\mathsf{D}_\mathcal{K}$, function exist_{TR} finds in polynomial time the required directed paths in J.

Finally, in steps 16–20, we check that a substitution π that maps the placeholder variables in $\mathcal{S}$ to the named individuals in I, and that maps all the other vertices of $\mathcal{S}$ to labelled nulls in I exists such that the relevant constraints expressed by the skeleton are satisfied. Using Figures 6.2 and 6.3, one can check that substitution

$$\pi = \{p_1 \mapsto b,\ y_1 \mapsto 3,\ y_2 \mapsto 1,\ y_3 \mapsto 4,\ y_4 \mapsto 5\}$$

satisfies the constraints imposed by $\mathcal{S}$; hence, $\mathsf{isSound}_{TR}$ returns t, indicating that candidate answer τ is sound.

6.2 Formalisation

We now formalise the intuitions we have just presented. In the rest of this section, we fix a consistent $\mathcal{ELHO}^+_\bot$ KB $\mathcal{K} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$, a conjunctive query q', and a candidate answer τ to q' over $\mathsf{D}_\mathcal{K}$. Finally, given a term $t \in \mathsf{term}(q')$, we let the canonical representative $\tau(t)_\approx$, and sets $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ and $\mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ be as specified in Definition 5.6 on page 57.

Due to equality rules, distinct constants can be identified by $\mathsf{D}_\mathcal{K}$; hence, to avoid dealing with equal constants while checking whether τ is sound, we replace in q' each term $t \in \mathsf{term}(q')$ that τ does not map to constants in $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ with the canonical representative $\tau(t)_\approx$; this replacement produces CQ q. Since q is obtained by replacing equals by equals, we have $\mathsf{D}_\mathcal{K} \models \tau(q)$; moreover, by Proposition 5.5, CQ q can be computed in time polynomial in the size of q and $\mathcal{K}$. Our filtering procedure uses q to check whether τ is sound.

Definition 6.1. *Conjunctive query q is obtained from q' by replacing each term $t \in \mathsf{term}(q')$ such that $\tau(t) \notin \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ with $\tau(t)_\approx$.*

Next, we define good and aux-simple binary atoms w.r.t. substitution τ.

Definition 6.2. *Let $R \in \mathsf{rol}_\mathcal{K}$ be a role and let $s, t \in \mathsf{term}(q)$ be terms. Atom $R(s,t)$ is* good *if at least one of the following conditions holds:*

- $\top_r \in \mathcal{L}(R)$,
- $\tau(t) \in N_{\mathfrak{I}}$, *or*
- $s = t$ *and* $\mathsf{D}_{\mathcal{K}} \models \mathbb{S}_R(\tau(s))$.

Atom $R(s,t)$ *is* aux-simple *if all of the following conditions hold:*

- $s \neq t$ *and* $\tau(t) \in \mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$,
- R *is a simple role, and*
- $\tau(s) = \tau(t)$ *implies that* $\mathsf{D}_{\mathcal{K}} \not\models \mathbb{S}_R(\tau(s))$.

Since $\mathcal{R}$ is a role hierarchy, for each role $R \in \mathsf{rol}_{\mathcal{R}}$, we have that $\top_r \in \mathcal{L}(R)$ if and only if $\top_r \sqsubseteq^*_{\mathcal{R}} R$; thus, by Proposition 5.5, we can then check whether $R(s,t)$ is good or aux-simple in time polynomial in $\mathcal{K}$. Note that, if $R(s,t)$ is not good, then t is a variable that τ maps to $\mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$. Furthermore, for each simple role $R \in \mathsf{rol}_{\mathcal{R}}$, we have that $\top_r \not\sqsubseteq^*_{\mathcal{R}} R$, and thus an atom cannot be at same time good and aux-simple. Moreover, each aux-simple atom $R(s,t)$ is such that R is simple and τ does not map the atom onto a self edge; hence, τ maps the atom onto a direct edge and, because direct edges propagate through simple role inclusions, we have that $\mathsf{D}_{\mathcal{K}} \models \mathbb{D}_R(\tau(s), \tau(t))$. Finally, if $\mathcal{K}$ is in $\mathcal{ELHO}_\bot$, each atom $R(s,t)$ is either good or aux-simple, as $\mathcal{K}$ contains neither self-restrictions, reflexive roles, nor transitive roles.

We next show how to compute the equality constraints for q by starting from the 'aux-simple forks' in q and by propagating the constraints further down the query. In addition, we define the query $q_\sim$ that is obtained by 'applying' the equality constraints to query q.

Definition 6.3. *Relation* $\sim \subseteq \mathsf{term}(q) \times \mathsf{term}(q)$ *for* q *and* τ *is the smallest reflexive, symmetric, and transitive relation closed under the* fork *rule.*

$$(\mathsf{fork})\ \frac{s' \sim t'}{s \sim t}\ R(s,s') \text{ and } P(t,t') \text{ are aux-simple atoms in } q \text{ w.r.t. } \tau$$

CQ $q_\sim$ *is obtained from* q *by replacing each term* $t \in \mathsf{term}(q)$ *with an arbitrary, but fixed representative of the equivalence class of* $\sim$ *containing* t.

To check whether $q_\sim$ is 'aux-acyclic', we next introduce the *connection graph* cg for q and τ that contains a set E_s of edges $\langle v, v' \rangle$ for each aux-simple atom $R(v, v') \in q_\sim$. In addition, cg also contains a set E_t of edges $\langle v, v' \rangle$ that we later use to reduce the nondeterminism in guessing a skeleton for $\sigma(q_\sim)$.

Definition 6.4. *The set Ψ_q of* placeholders *for q contains a fresh variable p_z uniquely associated with each variable $z \in \mathsf{var}(q_\sim)$. Then $\mathsf{cg} = \langle V, E_s, E_t \rangle$ is the* connection graph for q and τ *where $V = \mathsf{term}(q_\sim) \cup \Psi_q$ and $E_s, E_t \subseteq V \times V$ are the smallest sets of edges satisfying the following conditions.*

- *E_s contains $\langle s, t \rangle$ for all $s, t \in \mathsf{term}(q_\sim)$ for which a role R exists such that $R(s, t)$ is an aux-simple atom in $q_\sim$.*
- *E_t contains*
  - *$\langle p, z \rangle$ for each $p \in \Psi_q$ and each $z \in \mathsf{var}(q_\sim)$, and*
  - *$\langle s, t \rangle$ for all $s, t \in \mathsf{term}(q_\sim)$ for which roles $R_1, \ldots, R_n$ and constants $u_0, \ldots, u_n$ in $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ with $n > 0$ exist where $u_0 = \tau(s)$, $u_n = \tau(t)$, and $\mathsf{D}_\mathcal{K} \models \mathbb{D}_{R_i}(u_{i-1}, u_i)$ for each $i \in [1..n]$.*

By the definition of aux-simple atoms, we have that $E_s \subseteq E_t$; furthermore, each edge $\langle s, t \rangle \in E_t$ is such that $t \in \mathsf{var}(q_\sim)$ and $\tau(t) \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$.

Function $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ from Definition 6.5 ensures that τ satisfies the constraints in $\sim$, and that $q_\sim$ does not contain cycles consisting only of aux-simple atoms.

Definition 6.5. *Function $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ returns t if and only if the two following conditions hold.*

(1) For all $s, t \in \mathsf{term}(q)$, if $s \sim t$, then $\tau(s) = \tau(t)$.

(2) $\langle V, E_s \rangle$ is a directed acyclic graph.

We next define the notion of a variable renaming σ for q and τ. Condition (2) in Definition 6.6 ensures that such a variable renaming σ does not identify variables that cannot be mapped to the same element in I, whereas condition (4) ensures that $\sigma(q_\sim)$ does not contain aux-simple forks or cycles.

Definition 6.6. *A substitution σ is a* variable renaming for q and τ *if*

(1) $\mathsf{dom}(\sigma) = V \cap \mathsf{var}(q_\sim)$ *and* $\mathsf{rng}(\sigma) \subseteq V \cap \mathsf{var}(q_\sim)$,

(2) for each $v \in \mathsf{dom}(\sigma)$, *we have* $\tau(v) = \tau(\sigma(v))$,

(3) for each $v \in \mathsf{rng}(\sigma)$, *we have* $\sigma(v) = v$, *and*

(4) directed graph $\langle \sigma(V), \sigma(E_s) \rangle$ *is a forest.*

We next define the notion of a skeleton $\mathcal{S}$ for q and a variable renaming σ. We use the edges in $\sigma(E_t)$ to ensure that, for each edge $\langle v, v' \rangle \in \mathcal{S}$ with $v \notin \Psi_q$, a path from $\tau(v)$ to $\tau(v')$ in J exists that consists only of direct edges pointing to auxiliary constants.

Definition 6.7. *A* skeleton *for q and a variable renaming σ is a directed graph $\mathcal{S} = \langle \mathcal{V}, \mathcal{E} \rangle$ with $\mathcal{V} = \sigma(V)$ such that $\sigma(E_s) \subseteq \mathcal{E} \subseteq \sigma(E_t)$ and $\langle \mathcal{V}, \mathcal{E} \rangle$ is a forest whose roots are the individuals and the placeholders in $\mathcal{V}$.*

In Definition 6.8, we present function exist_{TR} that checks whether one can satisfy property (a) on page 88 for each edge $\langle v, v' \rangle \in \mathcal{S}$ that is labelled by a set L of roles.

Definition 6.8. *Given two constants u and u' from $\mathsf{D}_\mathcal{K}$, and a finite set of roles L, function $\mathsf{exist}_{TR}(u, u', L)$ returns* t *if and only if constants $u_1, \ldots, u_n$ in $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ with $n > 0$ and $u_n = u'$ exist where*

- *if $S \in L$ exists such that S is not transitive in $\mathcal{R}$, then $n = 1$, and*
- *$u_0 = u$ and $\mathsf{D}_\mathcal{K} \models \mathbb{D}_R(u_{i-1}, u_i)$ for each $R \in L$ and each $i \in [1..n]$.*

Finally, function check_{TR} in Definition 6.9 uses function exist_{TR} and entailment checking over $\mathsf{D}_\mathcal{K}$ to determine whether one can satisfy properties (a) and (b) for each vertex $v \in \mathcal{V}$.

Definition 6.9. *Given a skeleton vertex v, a constant u from $\mathsf{D}_\mathcal{K}$, and a function L that maps each skeleton edge to a finite set of roles and each skeleton root to a finite set of pairs of the form $\langle s, P \rangle$ where s is a skeleton vertex and P is a role, function $\mathsf{check}_{TR}(v, u, L)$ returns* t *if and only if the following two conditions hold.*

(a) Function $\mathsf{exist}_{TR}(u, \tau(v'), L(v, v'))$ returns t *for each skeleton edge $\langle v, v' \rangle \in \mathcal{S}$.*

Algorithm 1: $\mathsf{isSound}_{TR}(q, \mathsf{D}_\mathcal{K}, \tau)$

if $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau) = \mathsf{f}$ **then return** f
return t **if** each $R(s,t) \in q_\sim$ is good or aux-simple
guess a variable renaming σ for q and τ
guess a skeleton $\mathcal{S} = \langle \mathcal{V}, \mathcal{E} \rangle$ for q, σ, and τ
foreach $v \in \mathcal{V}$, **let** $L(v) = \emptyset$; **foreach** $\langle v, v' \rangle \in \mathcal{E}$, **let** $L(v, v') = \emptyset$
foreach *aux-simple atom* $R(s,t) \in \sigma(q_\sim)$, **add** R to $L(s,t)$
foreach *neither good nor aux-simple* $R(s,t) \in \sigma(q_\sim)$ **do**
    guess a role $P \in N_\mathfrak{R} \setminus \{\top_r, \bot_r\}$ with $\mathsf{D}_\mathcal{K} \models P(\tau(s), \tau(t))$ and $P \sqsubseteq^*_\mathcal{R} R$
    if $\langle s,t \rangle \notin \mathcal{E}$ *and* P *is not transitive* **then return** f
    if s *reaches* t *in* $\mathcal{S}$ **then**
        let $v_0, \ldots, v_n$ be the path from s to t in $\mathcal{S}$
    else
        let v_0 be the root that reaches t in $\mathcal{S}$ via $v_0, \ldots, v_n$
        add $\langle s, P \rangle$ to $L(v_0)$
    foreach $i \in [1..n]$, **add** P to $L(v_{i-1}, v_i)$
foreach $v \in \mathcal{V}$ **do**
    if $v \in \Psi_q$ **then**
        return f **if** no individual $a \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ exists such that $\mathsf{check}_{TR}(v, a, L) = \mathsf{t}$
    else
        return f **if** $\mathsf{check}_{TR}(v, \tau(v), L) = \mathsf{f}$
return t

(b) $\mathsf{D}_\mathcal{K} \models P(\tau(s), u)$ *holds for each pair* $\langle s, P \rangle \in L(v)$.

Theorem 6.10 shows that the candidate answer τ for q' over $\mathsf{D}_\mathcal{K}$ is sound if and only if the nondeterministic filtering procedure $\mathsf{isSound}_{TR}(q, \mathsf{D}_\mathcal{K}, \tau)$ from Algorithm 1 returns t. The proof of this result is given in Section 6.4.

Theorem 6.10. *For each substitution* π, $\mathcal{K} \models_{\mathcal{DL}} \pi(q')$ *if and only if a candidate answer* τ *to* q' *over* $\mathsf{D}_\mathcal{K}$ *exists such that* $\tau|_{\vec{x}} = \pi$ *and the following conditions hold:*

(1) for each $x \in \vec{x}$, $\tau(x) \in N_\mathfrak{I}$, *and*

(2) a nondeterministic computation exists such that $\mathsf{isSound}_{TR}(q, \mathsf{D}_\mathcal{K}, \tau)$ *returns* t.

Finally, we determine the complexity of the algorithm $\mathsf{isSound}_{TR}$. Towards this goal, we first determine the complexity of the auxiliary function isDSound.

Lemma 6.11. *Function* $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ *runs in time polynomial in the input size.*

Proof. We next show that conditions (1) and (2) in the definition of $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ can be implemented to run in polynomial time.

Condition (1). We argue that we can compute relation $\sim$ in time polynomial in the input size. More specifically, by Proposition 5.5 we can evaluate in polynomial time the precondition of the (fork) rule. In addition, the size of relation $\sim$ is bounded by $|\mathsf{term}(q)|^2$, the rules used to compute it are monotonic, and each inference can be applied in polynomial time, so the claim follows.

Condition (2). We first argue that we can compute the connection graph for q in polynomial time. Given terms s and t from $q_\sim$, we can test in polynomial time whether $\langle s, t\rangle \in E_t$ holds by checking whether $\tau(t)$ is reachable from $\tau(s)$ in the graph containing an edge $\langle u, u'\rangle$ for each $u \in \{\tau(s)\} \cup \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ and each $u' \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ for which a role R exists such that $\mathsf{D}_\mathcal{K} \models \mathbb{D}_R(u, u')$. Therefore, the connection graph $\mathsf{cg} = \langle V, E_s, E_t\rangle$ can be computed in polynomial time. We can check in linear time whether $\langle V, E_s\rangle$ is a acyclic by searching for a topological ordering of its vertices [21], so condition (2) in Definition 6.5 of function $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ can be implemented to run in polynomial time in the input size. □

The next lemma shows that function exist_{TR} can be decided in polynomial time as well.

Lemma 6.12. *Function* $\mathsf{exist}_{TR}(u, u', L)$ *runs in time polynomial in the input size.*

Proof. Let u and u' be constants and let L be a set of roles. We consider the two cases in Definition 6.8 separately.

If a role $S \in L$ exists that is not transitive, then we simply check whether $\mathsf{D}_\mathcal{K} \models \mathbb{D}_R(u, u')$ holds for each role $R \in L$. By Proposition 5.5, this check can be done in polynomial time.

Otherwise, if L contains only transitive roles, we check that u' is reachable from u in the directed graph that contains an edge $\langle c, c'\rangle$ for all constants $c \in \{u\} \cup \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ and $c' \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ such that $\mathsf{D}_\mathcal{K} \models \mathbb{D}_R(c, c')$ for each $R \in L$. By Proposition 5.5, such a graph can be computed in time polynomial in the size of $|\mathcal{K}|$, so all of these checks run in polynomial time. □

We are now ready to establish the complexity of the filtering procedure $\mathsf{isSound}_{TR}$; in Section 6.3 we will show that our function is worst-case optimal.

Theorem 6.13. *Function* $\mathsf{isSound}_{TR}(q, \mathsf{D}_\mathcal{K}, \tau)$ *can be implemented so that*

(1) it runs in nondeterministic polynomial time,

(2) if each binary atom in $q_\sim$ is either good or aux-simple w.r.t. τ, it runs in polynomial time, and

(3) if the query q is fixed, it runs in polynomial time in the size of $\mathcal{K}$.

Proof. By Lemma 6.11, the check in line 1 can be implemented to run in polynomial time. Hence, if each binary atom in $q_\sim$ is either good or aux-simple, function $\mathsf{isSound}_{TR}$ runs in time polynomial in the input size, and property (2) holds.

By Lemma 6.12 and Proposition 5.5, function check_{TR} can also be decided in polynomial time, and so the for-loop in lines 16–20 takes polynomial time as well. Since all other operations in lines 3–15 can clearly be implemented to run in nondeterministic polynomial time in the input size, property (1) holds.

For property (3), assume that the query q is fixed. Given that the number of variables occurring in q is fixed, the number of guessing steps required in lines 3 and 4 is fixed; also, the number of alternatives for these steps is linear in the size of $\mathcal{K}$. Thus, lines 3 and 4 require polynomial time. Moreover, the maximum number of iterations of the for-loop in lines 7–15 is fixed and the number of alternatives for the guessing steps in line 8 is linear in the size of $\mathcal{K}$. Thus, lines 7–15 require time polynomial in the size of $\mathcal{K}$. All other steps can be implemented in time polynomial in the size of $\mathcal{K}$, so property (3) holds. □

By Propositions 5.5 and 5.7, we can check whether an $\mathcal{ELHO}^+_\bot$ knowledge base is inconsistent using polynomial time; hence, by Theorem 6.10 we can check whether π is a certain answer to q over $\mathcal{K}$ using nondeterministic polynomial time in combined complexity (i.e., when the ABox, the RBox, the TBox, and the query are all part of the input), and in polynomial time in knowledge base complexity (i.e., when the query is fixed). Furthermore, when the query is not fixed, CQ answering is NP-hard already over relational databases [18]; and Calvanese et al. [13] showed that instance checking over $\mathcal{EL}$ knowledge bases is PTime-hard in data complexity. The following theorem summarises these results.

Theorem 6.14. *Given a substitution π, checking whether $\mathcal{K} \models_{\mathcal{DL}} \pi(q')$ is*

- PTime*-complete in data and KB complexities, and*
- NP*-complete in combined complexity.*

6.3 Lower Bound for Checking Candidate Answer Soundness

Even though our filtering procedure $\mathsf{isSound}_{TR}$ runs in nondeterministic polynomial time in the general case, Theorem 6.13 shows that the procedure runs in polynomial time if $\mathcal{K}$ does not contain self-restrictions, and transitive and reflexive roles. In Theorem 6.15, we show that this complexity increase is unavoidable: if $\mathcal{K}$ contains transitive roles, checking whether a candidate answer τ to q' over $\mathsf{D}_{\mathcal{K}}$ is sound is an NP-hard problem. We prove our claim by reducing the NP-hard problem of checking satisfiability of a 3CNF formula φ [33]. Towards this goal, we define an $\mathcal{ELHO}_{\bot}^{+}$ KB $\mathcal{K}_{\varphi}$ and a Boolean CQ q_{φ} such that φ is satisfiable if and only if $\mathcal{K}_{\varphi} \models_{\mathcal{DL}} q_{\varphi}$. Furthermore, we define a substitution τ_{φ} and show that τ_{φ} is a unique candidate answer to q_{φ} over the datalog program D_{φ} for $\mathcal{K}_{\varphi}$.

Theorem 6.15. *Checking whether a candidate answer is sound is* NP*-hard.*

Proof. The proof is by reduction from the NP-hard problem of checking the satisfiability of a 3CNF formula [33]. Let $\varphi = \bigwedge_{j=1}^{m} C_j$ be a 3CNF formula over variables $\{v_1, \ldots, v_n\}$, where each C_j is a set of three literals $C_j = \{l_{j,1}, l_{j,2}, l_{j,3}\}$. A sequence $\mu = l_{j_1,k_1}, \ldots, l_{j_\ell,k_\ell}$ of literals from φ is *consistent* if $\{v_i, \neg v_i\} \not\subseteq \mu$ for each $i \in [1..n]$. Such μ is a *truth sequence* for φ if for each $j \in [1..m]$, a literal $l \in \mu$ exists such that $l = l_{j,k}$ for some $k \in [1..3]$. Then φ is satisfiable if and only if there exists a consistent truth sequence for φ.

Let φ be a 3CNF formula. We proceed in three steps. We first define an $\mathcal{ELHO}_{\bot}^{+}$ KB $\mathcal{K}_{\varphi}$ and a Boolean CQ q_{φ} such that $\mathcal{K}_{\varphi} \models_{\mathcal{DL}} q_{\varphi}$ if and only if there exists a consistent truth sequence for φ. Then, we construct a substitution τ_{φ} and show that τ_{φ} is a candidate answer to q_{φ} over the datalog program D_{φ} for $\mathcal{K}_{\varphi}$. Finally, we show that τ_{φ} is the only candidate answer to τ_{φ}; thus $\mathcal{K}_{\varphi} \models_{\mathcal{DL}} q_{\varphi}$ if and only if τ_{φ} is sound.

Construction of $\mathcal{K}_{\varphi}$ and q_{φ}. In our construction of $\mathcal{K}_{\varphi}$, we use a fresh individual a_{ϵ}, fresh concepts A and G, fresh roles R and T, a fresh concept $L_{j,k}$ and a fresh role $S_{j,k}$ uniquely associated with each literal $l_{j,k}$, a fresh concept C_j uniquely associated with each clause C_j, and fresh roles P_i, N_i, and T_i uniquely associated with each variable v_i. Moreover, we associate each sequence $\mu = l_{j_1,k_1}, \ldots, l_{j_\ell,k_\ell}$ of literals from φ with the role chain $\rho_{\mu} = R \cdot S_{j_1,k_1} \cdot R \cdot S_{j_2,k_2} \cdots R \cdot S_{j_\ell,k_\ell} \cdot R$.

We will present our construction of $\mathcal{K}_\varphi$ in four stages, and at each stage we will describe how the additional axioms affect the universal interpretation I of $\mathcal{K}_\varphi$—that is, the possibly infinite set of assertions obtained by applying our chase variant to the rule base for $\mathcal{K}_\varphi$.

The first part of $\mathcal{K}_\varphi$ contains atom (6.3) and axioms (6.4)–(6.7). Then, for each sequence μ of literals from φ, a term w_μ exists such that $G(w_\mu) \in I$ and $\rho_\mu(a_\epsilon, w_\mu) \in I$.

$$A(a_\epsilon) \tag{6.3}$$

$$A \sqsubseteq \exists R.C_j \qquad \forall j \in [1..m] \tag{6.4}$$

$$A \sqsubseteq \exists R.G \tag{6.5}$$

$$C_j \sqsubseteq \exists S_{j,k}.L_{j,k} \qquad \forall j \in [1..m]\, \forall k \in [1..3] \tag{6.6}$$

$$L_{j,k} \sqsubseteq A \qquad \forall j \in [1..m]\, \forall k \in [1..3] \tag{6.7}$$

The second part of $\mathcal{K}_\varphi$ contains role inclusions (6.8)–(6.9). Consider an arbitrary literal $l_{j,k}$ and arbitrary terms w and w' such that $S_{j,k}(w, w') \in I$. Then, for each $i \in [1..n]$, we have that (a) $P_i(w, w') \in I$ if and only if the sequence of literals $v_i, l_{j,k}$ is consistent, and (b) $N_i(w, w') \in I$ if and only if the sequence of literals $\neg v_i, l_{j,k}$ is consistent.

$$S_{j,k} \sqsubseteq P_i \qquad \forall j \in [1..m]\, \forall k \in [1..3]\, \forall i \in [1..n] \text{ with } l_{j,k} \neq \neg v_i \tag{6.8}$$

$$S_{j,k} \sqsubseteq N_i \qquad \forall j \in [1..m]\, \forall k \in [1..3]\, \forall i \in [1..n] \text{ with } l_{j,k} \neq v_i \tag{6.9}$$

The third part of $\mathcal{K}_\varphi$ contains role inclusions (6.10)–(6.13). It should be clear that, for each sequence μ of literals from φ and each $i \in [1..n]$, we have that (c) $P_i(a_\epsilon, w_\mu) \in I$ if and only if the sequence of literals μ, v_i is consistent, and (d) $N_i(a_\epsilon, w_\mu) \in I$ if and only if the sequence of literals $\mu, \neg v_i$ is consistent.

$$R \sqsubseteq P_i \qquad \forall i \in [1..n] \tag{6.10}$$

$$R \sqsubseteq N_i \qquad \forall i \in [1..n] \tag{6.11}$$

$$P_i \cdot P_i \sqsubseteq P_i \qquad \forall i \in [1..n] \tag{6.12}$$

$$N_i \cdot N_i \sqsubseteq N_i \qquad \forall i \in [1..n] \tag{6.13}$$

The fourth part of $\mathcal{K}_\varphi$ contains role inclusions (6.14)–(6.16). It should be clear that, for each sequence μ of literals from φ and each $i \in [1..n]$, we have $T_i(a_\epsilon, w_\mu) \in I$ if and only if both v_i and $\neg v_i$ do not occur in μ. Moreover, for each $j \in [1..m]$, a term w_j exists such that $C_j(w_j) \in I$ and $T(w_j, w_\mu) \in I$ if and only if an index $k \in [1..3]$ exists such that $l_{j,k} \in \mu$.

$$P_i \sqsubseteq T_i \qquad \forall i \in [1..n] \tag{6.14}$$

$$N_i \sqsubseteq T_i \qquad \forall i \in [1..n] \tag{6.15}$$

$$T_i \sqsubseteq T \qquad \forall i \in [1..n] \tag{6.16}$$

Query q_φ is given in (6.17), where y and $\vec{z} = z_1, \ldots, z_m$ are fresh variables. Then $\mathcal{K}_\varphi \models_{\mathcal{DL}} q_\varphi$ if and only if a sequence μ of literals from φ exists such that, for each $i \in [1..n]$, we have $T_i(a_\epsilon, w_\mu) \in I$ and, for each $j \in [1..m]$, a term w_j exists such that $C_j(w_j) \in I$ and $T(w_j, w_\mu) \in I$. Hence, we have $\mathcal{K}_\varphi \models_{\mathcal{DL}} q_\varphi$ if and only if there exists a consistent truth sequence μ for φ.

$$q_\varphi = \exists y\, \exists \vec{z}.\, G(y) \wedge \bigwedge_{i=1}^{n} T_i(a_\epsilon, y) \wedge \bigwedge_{j=1}^{m} C_j(z_j) \wedge T(z_j, y) \tag{6.17}$$

Construction of candidate answer τ_φ. The datalog program D_φ for $\mathcal{K}_\varphi$ contains all the atoms and the translation of all axioms in $\mathcal{K}_\varphi$ into rules as specified in Table 5.1, apart from axioms (6.4)–(6.6) which are replaced by the datalog rules (6.18)–(6.20).

$$A(x) \rightarrow R(x, o_{R,C_j}) \wedge \mathbb{D}_R(x, o_{R,C_j}) \wedge C_j(o_{R,C_j}) \tag{6.18}$$

$$A(x) \rightarrow R(x, o_{R,G}) \wedge \mathbb{D}_R(x, o_{R,G}) \wedge G(o_{R,G}) \tag{6.19}$$

$$C_j(x) \rightarrow S_{j,k}(x, o_{S_{j,k},L_{j,k}}) \wedge \mathbb{D}_{S_{j,k}}(x, o_{S_{j,k},L_{j,k}}) \wedge L_{j,k}(o_{S_{j,k},L_{j,k}}) \tag{6.20}$$

Then let τ_φ be the substitution in (6.21).

$$\tau_\varphi(y) = o_{R,G} \qquad \tau_\varphi(z_j) = o_{R,C_j} \qquad \forall j \in [1..m] \tag{6.21}$$

We next show that $\mathsf{D}_\varphi \models \tau_\varphi(q_\varphi)$.

We first show that $\mathsf{D}_\varphi \models G(\tau_\varphi(y))$ and $\mathsf{D}_\varphi \models T_i(a_\epsilon, \tau_\varphi(y))$ for each $i \in [1..n]$. By atom (6.3) and by rule (6.19), we have $\mathsf{D}_\varphi \models R(a_\epsilon, \tau_\varphi(y))$ and $\mathsf{D}_\varphi \models G(\tau_\varphi(y))$. But then, for each $i \in [1..n]$, we have $\mathsf{D}_\varphi \models T_i(a_\epsilon, \tau_\varphi(y))$ due to axioms (6.10)–(6.11) and (6.14)–(6.15).

We next show that $\mathsf{D}_\varphi \models T(\tau_\varphi(z_j), \tau_\varphi(y))$ and $\mathsf{D}_\varphi \models C_j(\tau_\varphi(z_j))$ for each $j \in [1..m]$. Consider an arbitrary clause C_j; then, due to (6.3) and (6.18), we have $\mathsf{D}_\varphi \models C_j(\tau_\varphi(z_j))$. By rule (6.20) and axiom (6.7), for each literal $l_{j,k}$, an individual u exists such that $\mathsf{D}_\varphi \models S_{j,k}(\tau_\varphi(z_j), u) \wedge A(u)$. Then, by rule (6.19) we have $\mathsf{D}_\varphi \models R(u, \tau_\varphi(y))$; therefore, by axioms (6.10)–(6.16), we have that $\mathsf{D}_\varphi \models T(\tau_\varphi(z_j), \tau_\varphi(y))$, and so $\mathsf{D}_\varphi \models \tau_\varphi(q_\varphi)$.

Uniqueness of τ_φ. To prove the theorem, we are left to show that τ_φ is unique. Consider an arbitrary candidate answer ξ for q_φ and D_φ. Since $G(y)$ is an atom in q and only rule (6.19) derives atoms over G, we must have $\xi(y) = o_{R,G}$. Furthermore, for each $j \in [1..m]$, due to atom $C_j(z_j)$ in q and because only rule (6.18) derives assertions over concept C_j, we must have $\xi(z_j) = o_{R,C_j}$. Thus, $\xi = \tau_\varphi$, as required. □

This hardness result, however, is specific to our datalog encoding $\mathsf{D}_\mathcal{K}$ of $\mathcal{K}$, and we leave it open whether one can construct in polynomial time an interpretation J' of $\mathcal{K}$ such that the candidate answers to a CQ over J' can be filtered in polynomial time.

Recently, Gottlob et al. [40] showed that increasing the number of consequences captured by the datalog program $\mathsf{D}_\mathcal{K}$ is not the only way for achieving polynomial filtering. Assuming that each $\mathsf{D}_\mathcal{K}$ contains a small number of special constants not occurring in $\mathcal{K}$, we can extend the input query with a polynomial number of fresh existential variables ranging over the special constants in $\mathsf{D}_\mathcal{K}$ and encode the nondeterministic guesses in Algorithm 1 directly in the query. Hence, each candidate answer encapsulates the nondeterministic guesses required by our filtering procedure $\mathsf{isSound}_{TR}$; so filtering can be done in polynomial time. Although this solution would reduce the worst-case complexity of filtering, it is unlikely that it would prove beneficial in practice: the nondeterministic guesses are only 'moved' from the filtering to the query evaluation phase.

6.4 Proof of Correctness

In this section, we prove Theorem 6.10. We first prove that the filtering procedure $\mathsf{isSound}_{TR}$ from Algorithm 1 is sound, after which we prove that $\mathsf{isSound}_{TR}$ is complete.

Many of the auxiliary results that we will prove in this section will also be used in Chapter 9 to show the correctness of the filtering procedure for $\mathcal{ELRO}^{+}_{\bot}$ knowledge bases. For this reason, in the rest of this chapter, we fix a consistent $\mathcal{ELRO}^{+}_{\bot}$ knowledge base $\mathcal{K}$, a CQ q', and a candidate answer τ to q' over $\mathsf{D}_{\mathcal{K}}$. Furthermore, we let I and J be universal interpretations of $\Xi_{\mathcal{K}}$ and $\mathsf{D}_{\mathcal{K}}$, respectively, and we let ι be the function mapping the terms occurring in I to terms occurring in J as specified in Section 5.3; finally, we let q be the CQ obtained from q' as specified in Definition 6.1 on page 89. Unless otherwise stated, all the results that we prove in this section hold for $\mathcal{ELHO}^{+}_{\bot}$ as well as for $\mathcal{ELRO}^{+}_{\bot}$ KBs.

6.4.1 Soundness

Assume that $\tau(x) \in N_{\mathfrak{I}}$ for each $x \in \vec{x}$ and a nondeterministic computation exists such that $\mathsf{isSound}_{TR}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ returns t; we show that $\mathcal{K} \models_{\mathcal{DL}} \tau|_{\vec{x}}(q')$.

By Definition 5.8 of τ, we have that $\mathsf{D}_{\mathcal{K}} \models \tau(q)$; furthermore, by Theorem 2.3, we have that $\|\tau(q')\|_J \subseteq \mathsf{inst}_J$. Moreover, by Definition 6.1 of q, we have $\|\tau(q)\|_J \subseteq \mathsf{inst}_J$ and, for each term $t \in \mathsf{term}(q)$, we have $\|\tau(t)\|_J = \tau(t)$; thus $\tau(q) \subseteq \mathsf{inst}_J$. To prove the soundness claim, we will construct a ground substitution ν that maps the variables of q to terms in I such that $\nu(q) \subseteq \mathsf{inst}_I$ and ν satisfies the following notion of congruence.

Definition 6.16. *A substitution ν is* congruent *with τ if*

(C) $\iota(\nu(t)) = \tau(t)$ for each term $t \in \mathsf{term}(q)$.

The following lemma shows that finding such a substitution ν suffices to prove the soundness claim.

Lemma 6.17. *Each substitution ν that is congruent with τ satisfies the following three properties for all terms $s, t \in \mathsf{term}(q)$, each $A \in \mathsf{con}_{\mathcal{K}}$, and each $R \in \mathsf{rol}_{\mathcal{K}}$.*

(1) $A(\tau(s)) \in \mathsf{inst}_J$ implies that $A(\nu(s)) \in \mathsf{inst}_I$.

(2) $R(\tau(s), \tau(t)) \in \mathsf{inst}_J$ *and* $R(s,t)$ *is good imply that* $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$.

(3) $\nu(q) \subseteq \mathsf{inst}_I$ *implies that* $\mathcal{K} \models_{\mathcal{DL}} \tau|_{\vec{x}}(q')$.

Proof. Let ν be a substitution that is congruent with τ.

Property (1). Consider an atom $A(s)$ with $s \in \mathsf{term}(q)$, and assume that $A(\tau(s)) \in \mathsf{inst}_J$. By property (d1) of Lemma 5.13, for each term w with $\iota(w) = \tau(s)$, we have $A(w) \in \mathsf{inst}_I$. Since ν is congruent with τ, this holds in particular for $w = \nu(s)$, and so $A(\nu(s)) \in I$.

Property (2). Consider an atom $R(s,t)$ with $s, t \in \mathsf{term}(q)$, and assume that $R(s,t)$ is good and that $R(\tau(s), \tau(t)) \in \mathsf{inst}_J$ holds. By the definition of good atoms, one of the following holds.

- $\top_r \in \mathcal{L}(R)$ or $\tau(t) \in N_{\mathfrak{I}}$. By property (d3) of Lemma 5.13, for all terms w and w' with $\iota(w) = \tau(s)$ and $\iota(w') = \tau(t)$, we have that $R(w, w') \in \mathsf{inst}_I$. Since ν is congruent with τ, this holds in particular for $w = \nu(s)$ and $w' = \nu(t)$, and so $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$.
- $s = t$ and $\mathbb{S}_R(\tau(t)) \in \mathsf{inst}_J$. By property (d2) of Lemma 5.13, for each term w with $\iota(w) = \tau(t)$, we have that $R(w, w) \in \mathsf{inst}_I$. Since ν is congruent with τ, this holds in particular for $w = \nu(t)$, and so we also have that $R(\nu(t), \nu(t)) \in \mathsf{inst}_I$.

Property (3). Assume that $\nu(q) \subseteq \mathsf{inst}_I$; we show that a substitution π_* exists such that $\tau|_{\vec{x}} \subseteq \pi_*$ and $\|\pi_*(q')\|_I \subseteq \mathsf{inst}_I$. By Proposition 5.2, this shows that $\mathcal{K} \models_{\mathcal{DL}} \tau|_{\vec{x}}(q')$.

Let γ be the mapping from $\mathsf{term}(q')$ to $\mathsf{term}(q)$ such that q is obtained by replacing each $t \in \mathsf{term}(q')$ with $\gamma(t)$. Furthermore, let π_* be the substitution such that, for each term $t \in \mathsf{term}(q')$, we have $\pi_*(t) = \nu(t)$ if $\gamma(t) = t$, and otherwise, $\pi_*(z)$ is an arbitrary term w such that $\iota(w) = \tau(t)$. By the construction π_* and because ν is congruent with τ, for each term $t \in \mathsf{term}(q')$, we have $\iota(\pi_*(t)) = \tau(t)$. Since $\tau(x) \in N_{\mathfrak{I}}$ for each $x \in \vec{x}$ and ι is the identity on $N_{\mathfrak{I}}$, we have $\tau|_{\vec{x}} \subseteq \pi_*$.

We are left to show that $\|\pi_*(q')\|_I \subseteq \mathsf{inst}_I$. To this end, we show that, for each term $t \in \mathsf{term}(q')$, we have that $\|\pi_*(t)\|_I = \nu(\gamma(t))$. By the definition of π_*, the property holds for each term $t \in \mathsf{term}(q')$ with $\gamma(t) = t$. Consider an arbitrary term $t \in \mathsf{term}(q')$ such that $\gamma(t) \neq t$. By the definition of γ, we have $\gamma(t) = \|\tau(t)\|_J$ and $\gamma(t) \in N_{\mathfrak{I}}$. By Lemma 5.11, for each term w occurring in I with $\iota(w) = \tau(t)$, we have $\|w\|_I = \gamma(t)$. Because ν is

congruent with τ, this holds in particular for $w = \nu(t)$. By the definition of π_*, we then have $\|\pi_*(t)\|_I = \gamma(t) = \nu(\gamma(t))$. Since $\nu(q) \subseteq \mathsf{inst}_I$, we have $\|\pi_*(q')\|_I \subseteq \mathsf{inst}_I$. □

Next, we consider the case in which each binary atom in $q_\sim$ is either good or aux-simple, and we construct a substitution ν that is congruent with τ such that $\nu(q) \subseteq \mathsf{inst}_I$.

Soundness of isDSound

Assume that each atom $R(s,t) \in q_\sim$ is either good or aux-simple. Since $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ returns t, directed graph $\langle V, E_s \rangle$ is acyclic. Next, we show that $\langle V, E_s \rangle$ is also a forest, and later we construct ν by induction on $\langle V, E_s \rangle$.

Lemma 6.18. *Directed graph $\langle V, E_s \rangle$ is a forest.*

Proof. By Definition 6.4 of $\mathsf{cg} = \langle V, E_s, E_t \rangle$, E_s is a binary relation on $\mathsf{term}(q_\sim)$. Since $\langle V, E_s \rangle$ is acyclic, we are left to show that for each $t \in \mathsf{term}(q_\sim)$, at most one term s exists such that $\langle s, t \rangle \in E_s$. Assume the opposite; hence, terms s_1, s_2, and t exist in $\mathsf{term}(q_\sim)$ such that $s_1 \neq s_2$ and $\{\langle s_1, t \rangle, \langle s_2, t \rangle\} \subseteq E_s$. Then, roles R and P exist such that $R(s_1, t)$ and $P(s_2, t)$ are aux-simple atoms in $q_\sim$ and $\tau(t) \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$. By the definition of $\sim$, we have $s_1 \sim s_2$; and, by the construction of $q_\sim$, we have $s_1 = s_2$, which is a contradiction. □

We next construct substitution ν that is congruent with τ, and satisfies the two following properties.

(a) For all terms $s, t \in \mathsf{term}(q)$ with $s \sim t$, we have that $\nu(s) = \nu(t)$.

(b) For each aux-simple atom $R(v, v')$ of $q_\sim$, we have $R(\nu(v), \nu(v')) \in \mathsf{inst}_I$.

We define ν by structural induction on the forest $\langle V, E_s \rangle$; later we show that $\nu(q) \subseteq \mathsf{inst}_I$.

Base case. Consider a root $v \in V$. We distinguish two cases.

- $v \in V \setminus \Psi_q$, hence we have that $v \in \mathsf{term}(q_\sim)$. Fix an arbitrary term w such that $\iota(w) = \tau(v)$. By Lemmas 5.11 and 5.13, such a term w exists and $\|w\|_I = w$. For each term $s \in \mathsf{term}(q)$ with $s \sim v$, let $\nu(s) = w$. By condition (1) in Definition 6.5, we have $\tau(s) = \tau(v)$. Thus, property (C) in Definition 6.16, and property (a) hold.

- $v \in V \cap \Psi_q$. Let $\nu(v)$ be an arbitrary individual in $\mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$. As $v \in \Psi_q$, properties (C) in Definition 6.16, and (a) vacuously hold.

Inductive step. Consider an arbitrary $\langle v, v' \rangle \in E_s$ with $\nu(v)$ defined and $\nu(v')$ undefined and let R be an arbitrary role such that $R(v, v')$ is an aux-simple atom in $q_\sim$. By the definition of E_s, such a role R exists. Furthermore, by Definition 6.2 of aux-simple atoms, we have that $\mathbb{D}_R(\tau(v), \tau(v')) \in \mathsf{inst}_J$ and $\tau(v')$ is of the form $o_{P,B}$. Moreover, we have that $\{v, v'\} \subseteq \mathsf{term}(q_\sim) \subseteq \mathsf{term}(q)$. By property (d5) of Lemma 5.13, for each term w with $\iota(w) = \tau(v)$, a term w^* exists such that $R(w, w^*) \in \mathsf{inst}_I$ and $\iota(w^*) = \tau(v')$. Since ν is congruent with τ, this holds in particular for $w = \nu(v)$. Then, for each term $s \in \mathsf{term}(q)$ with $s \sim v'$, let $\nu(s) = w^*$. Properties (a) and (b) immediately hold. By condition (1) in Definition 6.5, we have $\tau(s) = \tau(v)$, and so property (C) in Definition 6.16 holds as well.

Lemma 6.19. *Substitution ν satisfies $\nu(q) \subseteq \mathsf{inst}_I$.*

Proof. We show that $\nu(q) \subseteq \mathsf{inst}_I$ holds by considering the various atoms of q.

Consider an atom $A(s)$ in q. Because ν is congruent with τ, by property (1) of Lemma 6.17 we have that $A(\nu(s)) \in I$.

Consider an atom $R(s', t')$ in q. By Definition 6.3 of $q_\sim$, an atom $R(s, t)$ occurs in $q_\sim$ such that $s' \sim s$ and $t' \sim t$. By condition (1) in Definition 6.5, we have $\tau(s') = \tau(s)$ and $\tau(t') = \tau(t)$. By assumption, we have $R(\tau(s'), \tau(t')) \in \mathsf{inst}_J$ and so $R(\tau(s), \tau(t)) \in \mathsf{inst}_J$ as well. By property (a) of ν, it suffices to show that $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$. Given that our algorithm returns t in line 2, one of the following holds.

- $R(s, t)$ is good. Because ν is congruent with τ, by property (2) of Lemma 6.17 we have $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$.
- $R(s, t)$ is aux-simple. By property (a) of ν, we have that $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$. □

Then, by property (3) of Lemma 6.17 we obtain the following result that holds even when $\mathcal{K}$ is in $\mathcal{ELRO}^+_\bot$.

Lemma 6.20. *If* $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ *returns* t *and each binary atom of $q_\sim$ is either good or aux-simple, then $\mathcal{K} \models_{\mathcal{DL}} \tau|_{\vec{x}}(q')$.*

Soundness of $\mathsf{isSound}_{TR}$

We are left to construct substitution ν in case $q_\sim$ contains a binary atom that is neither good nor aux-simple, and so $\mathsf{isSound}_{TR}$ returns t in line 21. For the rest of this soundness proof, we assume that $\mathcal{K}$ is in $\mathcal{ELHO}^+_\bot$.

We next construct substitution ν that is congruent with τ, and satisfies the following properties.

(i) For all terms $s, t \in \mathsf{term}(q)$ such that $s \sim t$ or $\sigma(s) = t$, we have that $\nu(s) = \nu(t)$.

(ii) For each edge $\langle v, v' \rangle \in \mathcal{E}$ and each role $R \in L(v, v')$, we have $R(\nu(v), \nu(v')) \in \mathsf{inst}_I$.

By the definition of a skeleton for q and σ, graph $\mathcal{S}$ is a forest rooted in the individuals and the placeholders occurring in $\mathcal{V}$. Next, we define ν by structural induction on $\mathcal{S}$; later we show that $\nu(q) \subseteq \mathsf{inst}_I$.

Base case. Consider a root $v \in \mathcal{V}$. We distinguish two cases.

- $v \in \mathcal{V} \setminus \Psi_q$ and therefore $v \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$. Given that each element in $\mathsf{rng}(\sigma)$ is a variable, no term $s \in \mathsf{term}(q)$ exists such that $\sigma(s) = v$. Then, for each term $s \in \mathsf{term}(q)$ with $s \sim v$, let $\nu(s) = v$. By condition (1) in Definition 6.5, we have $\tau(s) = \tau(v)$. Thus, properties (C) in Definition 6.16, and (i) are satisfied.

- $v \in \mathcal{V} \cap \Psi_q$. By the definitions of relation $\sim$ and variable renaming σ, no term $s \in \mathsf{term}(q)$ exists such that $s \sim v$ or $\sigma(s) = v$. Then let $\nu(v)$ be an arbitrary individual $a \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ such that function $\mathsf{check}_{TR}(v, a, L)$ returns t. Because the condition in line 18 is not satisfied, such an individual exists. Properties (C) in Definition 6.16 and (i) vacuously hold.

Inductive step. Consider $\langle v, v' \rangle \in \mathcal{E}$ such that $\nu(v)$ has been defined, but $\nu(v')$ has not. Let $w_0 = \nu(v)$; and let $u_0 = \tau(v)$, if $v \notin \Psi_q$, otherwise let $u_0 = \nu(v)$. Note that, in either case, we have that $\iota(w_0) = u_0$. Because the checks in lines 18 and 20 fail, function $\mathsf{exist}_{TR}(u_0, \tau(v'), L(v, v'))$ returns t. Hence, auxiliary constants $u_1, \ldots, u_n$ with $n > 0$ and $u_n = \tau(v')$ exist in $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ such that, for each $i \in [1..n]$ and each $R \in L(v, v')$, we have $\mathbb{D}_R(u_{i-1}, u_i) \in \mathsf{inst}_J$. By property (d5) of Lemma 5.13, for each $i \in [1..n]$, a term w_i exists

such that $\iota(w_i) = u_i$ and, for each $R \in L(v, v')$, we have $R(w_{i-1}, w_i) \in \mathsf{inst}_I$. Then, for each term $s \in \mathsf{term}(q)$ with $s \sim v'$ or $\sigma(s) = v'$, let $\nu(s) = w_n$. Property (i) is clearly satisfied. Property (C) in Definition 6.16 is also satisfied, since $\sigma(s) = v'$ implies that $\tau(s) = \tau(v')$, by construction of σ, and $s \sim v'$ implies that $\tau(s) = \tau(v')$, by condition (1) in Definition 6.5. For property (ii) we distinguish two cases.

- A role $S \in L(v, v')$ exists such that S is not transitive in $\mathcal{R}$. By Definition 6.8 of function exist_{TR}, we have that $n = 1$. Therefore, we have that $\nu(v) = w_0$ and $\nu(v') = w_1$, and $R(\nu(v), \nu(v')) \in \mathsf{inst}_I$ for each $R \in L(v, v')$.
- For each role $R \in L(v, v')$, we have that R is transitive in $\mathcal{R}$. Since no rule is applicable to I and $R(w_{i-1}, w_i) \in \mathsf{inst}_I$ for each $i \in [1..n]$, we have that $R(w_0, w_n) \in \mathsf{inst}_I$, and so $R(\nu(v), \nu(v')) \in \mathsf{inst}_I$.

Lemma 6.21. *Substitution ν satisfies $\nu(q) \subseteq \mathsf{inst}_I$.*

Proof. We next show that $\nu(q) \subseteq \mathsf{inst}_I$ holds by considering the atoms occurring in q.

Consider an atom $A(s)$ in q. Because ν is congruent with τ, by property (1) of Lemma 6.17 we have $A(\nu(s)) \in I$.

Consider an atom $R(s', t')$ in q. By Definition 6.3 of $q_\sim$, terms s'' and t'' occur in $q_\sim$ such that $s' \sim s''$, $t' \sim t''$, and $R(s'', t'')$ is an atom in $q_\sim$. By condition (1) in Definition 6.5, we have $\tau(s') = \tau(s'')$ and $\tau(t') = \tau(t'')$. Therefore, $R(\tau(s''), \tau(t'')) \in \mathsf{inst}_J$. By definition of $\sigma(q_\sim)$, terms s and t occur in $\sigma(q_\sim)$ such that $\sigma(s'') = s$, $\sigma(t'') = t$, and $R(s, t)$ is an atom in $\sigma(q_\sim)$. By Definition 6.6 of σ, we have $\tau(t'') = \tau(t)$ and $\tau(s'') = \tau(s)$; and so $R(\tau(s), \tau(t)) \in \mathsf{inst}_J$ as well. By property (i) in the definition of ν, it suffices to show that $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$. Towards this goal, we consider three distinct cases.

$R(s, t)$ is good. Because ν is congruent with τ, by property (2) of Lemma 6.17 we have $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$.

$R(s, t)$ is aux-simple. By the definition of $\mathcal{E}$, we have $\langle s, t \rangle \in \mathcal{E}$; moreover, by line 6, we also have $R \in L(s, t)$. By property (ii) in the definition of ν, we have $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$.

$R(s, t)$ is neither good nor aux-simple. Let P and $v_0, \ldots, v_n$ be as determined in steps 8–15 when Algorithm 1 considers atom $R(s, t)$. Please note that $v_n = t$. By line 8 we

have that $P \sqsubseteq^*_{\mathcal{R}} R$, whereas by line 15, for each $i \in [1..n]$, we have $P \in L(v_{i-1}, v_i)$; thus, by property (ii), we have $P(\nu(v_{i-1}), \nu(v_i)) \in \mathsf{inst}_I$. Due to the check in line 9, one of the following holds.

- $\langle s, t\rangle \in \mathcal{E}$. Then, we have that $n = 1$, $v_0 = s$, and $v_1 = t$; thus $P(\nu(s), \nu(t)) \in \mathsf{inst}_I$.
- s reaches t in $\mathcal{E}$ and P is transitive. Since I is closed under $\Xi_{\mathcal{K}}$, $v_0 = s$, and $v_n = t$, we have $P(\nu(s), \nu(t)) \in \mathsf{inst}_I$.
- s does not reach t in $\mathcal{E}$ and P is transitive. Therefore, v_0 is the root that reaches t in $\mathcal{S}$ and $\langle s, P\rangle \in L(v_0)$. Since I is closed under $\Xi_{\mathcal{K}}$, we have $P(\nu(v_0), \nu(t)) \in \mathsf{inst}_I$. We next show that $P(\nu(s), \nu(v_0)) \in \mathsf{inst}_I$, which shows that $P(\nu(s), \nu(t)) \in \mathsf{inst}_I$, by considering two alternative cases.
  - $v_0 \in \mathcal{V} \setminus \Psi_q$ and therefore v_0 is an individual from $N_{\mathfrak{I}}$. Because the condition in line 20 is not satisfied, function $\mathsf{check}_{TR}(v_0, \tau(v_0), L)$ returns t. Then, by Definition 6.9 of function check_{TR}, we have that $P(\tau(s), \tau(v_0)) \in \mathsf{inst}_J$. Because ν is congruent with τ, by property (d3) of Lemma 5.13, for each term w with $\iota(w) = \tau(s)$, we have that $P(w, \nu(v_0)) \in \mathsf{inst}_I$. In particular, this holds for $w = \nu(s)$. Since $v_0 \in N_{\mathfrak{I}}$, we have $\nu(v_0) = \tau(v_0)$, and so $P(\nu(s), \nu(v_0)) \in \mathsf{inst}_I$.
  - $v_0 \in \mathcal{V} \cap \Psi_q$. By the construction of ν, we have $\nu(v_0) = a$ for some individual $a \in \mathsf{r\text{-}ind}_{\mathsf{D}_{\mathcal{K}}}$ such that function $\mathsf{check}_{TR}(v_0, a, L)$ returns t. Then, by Definition 6.9 of function check_{TR}, we have that $P(\tau(s), a) \in \mathsf{inst}_J$. Because ν is congruent with τ, by property (d3) of Lemma 5.13, for each term w with $\iota(w) = \tau(s)$, we have that $P(w, a) \in \mathsf{inst}_I$. In particular, this holds for $w = \nu(s)$. Because $\nu(v_0) = a$, we then have $P(\nu(s), \nu(v_0)) \in \mathsf{inst}_I$.

In all these cases, we have that $P(\nu(s), \nu(t)) \in \mathsf{inst}_I$. Since $P \sqsubseteq^*_{\mathcal{R}} R$ and I is closed under $\Xi_{\mathcal{K}}$, we have that $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$. □

By property (2) of Lemma 6.17, the above lemma shows that our filtering function $\mathsf{isSound}_{TR}$ is correct when $\mathcal{K}$ is in $\mathcal{ELHO}^+_\bot$.

6.4.2 Completeness

Let π be a substitution where $\mathsf{dom}(\pi) = \vec{x}$ and each element in $\mathsf{rng}(\pi)$ is an individual from $\mathsf{ind}_{\mathcal{K}}$, and assume that $\mathcal{K} \models_{\mathcal{DL}} \pi(q')$; we show that a candidate answer τ to q' over $\mathsf{D}_{\mathcal{K}}$ and a nondeterministic computation exist such that $\pi \subseteq \tau$ and $\mathsf{isSound}_{TR}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ returns t. In the following, let dir_I be the relation over the terms occurring in inst_I defined just before Lemma 5.3 on page 55. Recall that this relation contains each edge $\langle w, w' \rangle$ with $w' \in \Phi_N$ occurring in I that is direct for some role R.

Because $\mathcal{K}$ is consistent, by Proposition 5.2, we have that $\Xi_{\mathcal{K}} \models \pi(q')$; furthermore, by Theorem 2.3, a substitution π_* with $\mathsf{dom}(\pi_*) = \mathsf{var}(q')$ exists such that $\pi \subseteq \pi_*$ and $\|\pi_*(q')\|_I \subseteq \mathsf{inst}_I$. We next define substitution τ.

Definition 6.22. *Substitution τ is defined as follows.*

$$\tau(z) = \iota(\pi_*(z)) \qquad \forall z \in \mathsf{var}(q') \tag{6.22}$$

Please note that, because ι is the identity on $N_{\mathfrak{I}}$ and $\pi_*(x) \in \mathsf{ind}_{\mathcal{K}}$ for each $x \in \vec{x}$, we have that $\tau(x) = \pi_*(x) = \pi(x)$. By Lemmas 5.11 and 5.12, we have $\|\tau(q')\|_J \subseteq \mathsf{inst}_J$, and so $\mathsf{D}_{\mathcal{K}} \models \tau(q')$; that is, τ is a candidate answer to q' over $\mathsf{D}_{\mathcal{K}}$. Furthermore, by Definition 6.1 of q, we have $\tau(q) \subseteq \mathsf{inst}_J$ and, since $\tau(t) = \iota(\pi_*(t))$ for each term $t \in \mathsf{term}(q')$, we have $\pi_*(q) \subseteq \mathsf{inst}_I$ due to Lemma 5.11.

We are left to show that a nondeterministic computation exist such that filtering function $\mathsf{isSound}_{TR}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ returns t. To this end, Lemma 6.23 shows that π_* maps each aux-simple atom to a direct edge in I, while Lemma 6.24 shows that $\mathsf{isDSound}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ returns t.

Lemma 6.23. *For each role $R \in \mathsf{rol}_{\mathcal{K}}$ and all terms $s, t \in \mathsf{term}(q)$, $R(\pi_*(s), \pi_*(t)) \in \mathsf{inst}_I$ and $R(s, t)$ is aux-simple imply that $\langle \pi_*(s), \pi_*(t) \rangle \in \mathsf{dir}_I$.*

Proof. Consider an arbitrary aux-simple atom $R(s, t)$ such that $R(\pi_*(s), \pi_*(t)) \in \mathsf{inst}_I$. Then, we have that role R is simple, $s \neq t$ and $\tau(t) \in \mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$, and $\tau(s) = \tau(t)$ implies that $\mathbb{S}_R(\tau(t)) \notin \mathsf{inst}_J$. By Lemma 5.12 and the definition of τ, we have $\pi_*(t) \in \Phi_N$ and $\pi_*(s) = \pi_*(t)$ implies that $\mathbb{S}_R(\pi_*(t)) \notin I$. By property (3) of Lemma 5.3, we have that $\pi_*(s) \neq \pi_*(t)$. But then, by property (4) of of Lemma 5.3, we have $\langle \pi_*(s), \pi_*(t) \rangle \in \mathsf{dir}_I$. □

Lemma 6.24. *Function* $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ *returns* t *and* $\pi_*(q_\sim) \subseteq \mathsf{inst}_I$.

Proof. We show that the two conditions of Definition 6.5 are satisfied.

Condition (1). We prove that, for each $s \sim t$, we have $\tau(s) = \tau(t)$ and $\pi_*(s) = \pi_*(t)$. Note that, by the definition of $q_\sim$, this also implies that $\pi_*(q_\sim) \subseteq \mathsf{inst}_I$. We proceed by induction on the number of steps required to derive $s \sim t$. For the base case, the empty relation $\sim$ clearly satisfies the two properties. For the inductive step, consider an arbitrary relation $\sim$ obtained in n steps that satisfies these constraints; we show that the same holds for all constraints derivable from $\sim$. We focus on the (fork) rule, as the derivation of $s \sim t$ due to reflexivity, symmetry, or transitivity clearly preserves the required properties. Let s'_1, s_2, s'_2, and s_2 be arbitrary terms in $\mathsf{term}(q)$ such that $s'_1 \sim s'_2$ is obtained in n steps and let R_1 and R_2 be arbitrary roles such that atoms $R_1(s_1, s'_1)$ and $R_2(s_2, s'_2)$ occur in q and are aux-simple. By the inductive hypothesis, we have $\tau(s'_1) = \tau(s'_2)$ and $\pi_*(s'_1) = \pi_*(s'_2)$. Due to $\pi_*(q) \subseteq \mathsf{inst}_I$, by Lemma 6.23, we have $\{\langle \pi_*(s_1), \pi_*(s'_1)\rangle, \langle \pi_*(s_2), \pi_*(s'_2)\rangle\} \subseteq \mathsf{dir}_I$. As $\pi_*(s'_1) = \pi_*(s'_2)$ and because property (1) of Lemma 5.3 states that dir_I is a forest rooted in $N_\mathfrak{I}$, we have $\pi_*(s_1) = \pi_*(s_2)$. By the construction of τ, we finally have that $\tau(s_1) = \tau(s_2)$.

Condition (2). We show that graph $\langle V, E_s\rangle$ is acyclic. Assume the opposite; hence, vertices $v_0, \dots, v_m$ with $v_m = v_0$ exist in V such that $m > 0$ and $\langle v_{i-1}, v_i\rangle \in E_s$ for each $i \in [1..m]$. By the definition of E_s, for each $i \in [0..m]$, we have that $\tau(v_i) \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$. Consider an arbitrary $i \in [1..m]$ and the edge $\langle v_{i-1}, v_i\rangle \in E_s$. By the definition of E_s, a role R_i exists such that $R_i(v_{i-1}, v_i)$ is an aux-simple atom in $q_\sim$. Due to $\pi_*(q_\sim) \subseteq \mathsf{inst}_I$, by Lemma 6.23, we have $\langle \pi_*(v_{i-1}), \pi_*(v_i)\rangle \in \mathsf{dir}_I$. As $v_m = v_0$, relation dir_I is not a forest, which contradicts property (1) of Lemma 5.3. □

We are left to show that $\mathsf{isSound}_{TR}(q, \mathsf{D}_\mathcal{K}, \tau)$ returns t. To this end, we first define the variable renaming σ for q and τ, and the skeleton $\mathcal{S} = \langle \mathcal{V}, \mathcal{E}\rangle$ for q, σ, and τ.

Definition 6.25. *For V the vertices of the connection graph for q and τ, substitution σ maps each variable $z \in V \cap \mathsf{var}(q_\sim)$ to an arbitrary, but fixed variable $z' \in V \cap \mathsf{var}(q_\sim)$ such that $\pi_*(z) = \pi_*(z')$.*

Then, for all terms $s, t \in \mathsf{term}(\sigma(q_\sim))$ such that $s \neq t$, we have that $\pi_*(s) \neq \pi_*(t)$. Also,

since $\pi_*(q_\sim) \subseteq \mathsf{inst}_I$, we have that $\pi_*(\sigma(q_\sim)) \subseteq \mathsf{inst}_I$.

Definition 6.26. *Let $\mathcal{V} = \sigma(V)$, let dir_I^+ be the transitive closure of dir_I, and let $I_\mathcal{V}$ be the set containing each individual $a \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ for which a variable $z \in \mathsf{var}(\sigma(q_\sim))$ exists such that $a \notin \mathcal{V}$ and $\langle a, \pi_*(z)\rangle \in \mathsf{dir}_I^+$. Let ν be an arbitrary, but smallest substitution such that $\pi_* \subseteq \nu$ and, for each individual $a \in I_\mathcal{V}$, a unique placeholder $p \in \Psi_q$ exists such that $\nu(p) = a$. The skeleton is given by $\mathcal{S} = \langle \mathcal{V}, \mathcal{E}\rangle$ where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the smallest relation containing $\langle v, v'\rangle \in \mathcal{E}$ for all vertices $v, v' \in \mathcal{V}$ such that $\langle \nu(v), \nu(v')\rangle \in \mathsf{dir}_I^+$ and no vertex $v'' \in \mathcal{V}$ exists such that $\langle \nu(v), \nu(v'')\rangle \in \mathsf{dir}_I^+$ and $\langle \nu(v''), \nu(v')\rangle \in \mathsf{dir}_I^+$.*

Since $\pi_* \subseteq \nu$, we clearly have that $\nu(\sigma(q_\sim)) \subseteq \mathsf{inst}_I$. Furthermore, consider an arbitrary edge $\langle v, v'\rangle \in \mathcal{E}$. Due to $\langle \nu(v), \nu(v')\rangle \in \mathsf{dir}_I^+$, we have that terms $w_0, \dots, w_k$, roles $R_1, \dots, R_k$, and concepts $A_1, \dots, A_k$ exist such that $w_0 = \nu(v)$, $w_k = \nu(v')$, and for each $i \in [1..k]$ we have that w_i's type is R_i, A_i and $\mathbb{D}_R(w_{i-1}, w_i) \in \mathsf{inst}_I$. Note that all of these are uniquely defined by the edge.

Next, we show that σ and $\mathcal{S}$ are indeed a variable renaming and a skeleton, respectively.

Lemma 6.27. *Substitution σ is a variable renaming for q and τ, and directed graph $\mathcal{S}$ is a skeleton for q, σ, and τ.*

Proof. We first prove that $\mathcal{S}$ is a skeleton, later we show that σ is a variable renaming.

$\mathcal{S}$ is a skeleton. By property (1) of Lemma 5.3, relation dir_I is a forest, so the graph $\mathcal{S} = \langle \mathcal{V}, \mathcal{E}\rangle$ is a forest rooted in the individuals and the placeholders occurring in $\mathcal{V}$. We are left to prove that $\sigma(E_s) \subseteq \mathcal{E} \subseteq \sigma(E_t)$.

We first show that $\sigma(E_s) \subseteq \mathcal{E}$. Consider an edge $\langle v, v'\rangle \in \sigma(E_s)$. By the definition of E_s, an aux-simple atom $R(s, t) \in q_\sim$ exists such that $\sigma(s) = v$ and $\sigma(t) = v'$. By Lemma 6.24, we have that $\pi_*(q_\sim) \subseteq \mathsf{inst}_I$; furthermore, by Lemma 6.23, we have that $\langle \pi_*(s), \pi_*(t)\rangle \in \mathsf{dir}_I$. By the definition of σ and ν, we have that $\pi_*(s) = \pi_*(v) = \nu(v)$, and $\pi_*(t) = \pi_*(v') = \nu(v')$. Thus, $\langle v, v'\rangle \in \mathcal{E}$.

We next show that $\mathcal{E} \subseteq \sigma(E_t)$. Consider an edge $\langle v, v'\rangle \in \mathcal{E}$. By the definition of $\mathcal{E}$, we have that $\langle \nu(v), \nu(v')\rangle \in \mathsf{dir}_I^+$ and $v' \in \mathsf{var}(\sigma(q_\sim))$. If $v \in \Psi_q$, then $\langle v, v'\rangle \in \sigma(E_t)$ by the definition of E_t; so in the following we consider the case in which $v \notin \Psi_q$. Let $w_0, \dots, w_k$ be

the terms, let $R_1, \dots, R_k$ be the roles, and let $A_1, \dots, A_k$ be the concepts uniquely associated with $\langle v, v' \rangle \in \mathcal{E}$. By Lemma 5.12, we have that $\mathbb{D}_{R_i}(\iota(w_{i-1}), \iota(w_i)) \in J$ and $\iota(w_i) \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ for each $i \in [1..k]$. By the definition of τ, we also have that $\iota(w_0) = \tau(v)$ and $\iota(w_k) = \tau(v')$. Thus, $\langle v, v' \rangle \in \sigma(E_t)$.

σ is a variable renaming. Clearly, σ satisfies conditions (1)–(3) in Definition 6.6. We are left to show that σ satisfies condition (4) in the definition. Since $\sigma(E_s) \subseteq \mathcal{E}$, $\mathcal{V} = \sigma(V)$, and $\langle \mathcal{V}, \mathcal{E} \rangle$ is a forest, graph $\langle \sigma(V), \sigma(E_s) \rangle$ is a forest as well, as required. □

Finally, we are ready to prove completeness.

Lemma 6.28. *If $\mathcal{K}$ is in $\mathcal{ELHO}^+_\bot$, then a nondeterministic computation exists such that filtering procedure $\mathsf{isSound}_{TR}(q, \mathsf{D}_\mathcal{K}, \tau)$ returns t.*

Proof. By Lemma 6.24, the condition in step 1 in Algorithm 1 is not satisfied; furthermore, we have that $\pi_*(q_\sim) \subseteq \mathsf{inst}_I$. If each binary atom $R(s, t)$ occurring in $q_\sim$ is either good or aux-simple, then our algorithm returns t in step 2; hence, in the rest of this proof, we assume that this is not the case.

Let variable renaming σ be as specified in Definition 6.25, and let substitution ν and skeleton $\mathcal{S} = \langle \mathcal{V}, \mathcal{E} \rangle$ be as specified in Definition 6.26.

Consider an edge $\langle v, v' \rangle \in \mathcal{E}$, and let $w_0, \dots, w_k$ be the terms, let $R_1, \dots, R_k$ be the roles, and let $A_1, \dots, A_k$ be the concepts uniquely associated with the edge $\langle v, v' \rangle$. Then a role P is exist_{TR}*-compatible* with the edge $\langle v, v' \rangle$ if

- if P is not transitive in $\mathcal{R}$, then $k = 1$, and
- for each $i \in [1..k]$, we have $\mathbb{D}_P(\iota(w_{i-1}), \iota(w_i)) \in \mathsf{inst}_J$.

To prove the lemma, we will show that the following two properties hold for each $v \in \mathcal{V}$.

($\diamondsuit$) For each edge $\langle v, v' \rangle \in \mathcal{E}$ and each role $P \in L(v, v')$, we have that P is exist_{TR}-compatible with the edge $\langle v, v' \rangle$.

($\heartsuit$) For each pair $\langle s, P \rangle \in L(v)$, we have $P(\tau(s), \nu(v)) \in \mathsf{inst}_J$.

Please note that, by lines 13–14 in Algorithm 1, we have that $\langle s, P\rangle \in L(v)$ implies that v is a root of $\mathcal{S}$. Hence, either $v \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ and $\tau(v) = \nu(v)$, or $v \in \Psi_q$ and $\nu(v) \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$. Therefore, by the definition of function exist_{TR} (Definition 6.8) and the above definition of exist_{TR}-compatibility, properties ($\heartsuit$) and ($\diamondsuit$) imply that

- for each $v \in \mathcal{V} \cap \Psi_q$, we have that $\mathsf{check}_{TR}(v, \nu(v), L)$ returns t in line 18, and
- for each $v \in \mathcal{V} \setminus \Psi_q$, we have that $\mathsf{check}_{TR}(v, \tau(v), L)$ returns t in line 20.

For the loop in line 6, consider an arbitrary atom $R(s,t)$ in $\sigma(q_\sim)$ that is aux-simple. By Definition 6.2 of aux-simple atom, we have $\mathbb{D}_R(\tau(s), \tau(t)) \in \mathsf{inst}_J$. By the definition of τ, we have $\iota(\nu(s)) = \tau(s)$ and $\iota(\nu(t)) = \tau(t)$, so role R is compatible with the edge $\langle s, t\rangle$.

For the loop in lines 7–15, consider an arbitrary atom $R(s,t)$ in $\sigma(q_\sim)$ that is neither good nor aux-simple. We next determine the nondeterministic choices that preserve ($\heartsuit$) in line 14, ($\diamondsuit$) in line 15, and that satisfy the conditions in lines 8 and 9. By assumption, we have $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$; so a non-empty role chain $\rho = \chi_0 \cdot R_1 \cdot \chi_1 \cdots R_m \cdot \chi_m$ with $\rho \in \mathcal{L}(R)$ and terms $w_0, \ldots, w_m$ with $w_0 = \nu(s)$ and $w_m = \nu(t)$ exist that satisfy property (2) of Lemma 5.3. Since $\mathcal{R}$ is a role hierarchy, for each role T occurring in $\rho \in \mathcal{L}(R)$, we have that $T \sqsubseteq^*_\mathcal{R} R$.

We next show that $m \geq 1$. Assume that $m = 0$; hence, we have that $\nu(s) = \nu(t)$ and $\rho = \chi_0$. By the definition of variable renaming σ, we have that $s = t$. Furthermore, ρ is not empty so χ_0 contains at least one role S such that $\mathbb{S}_S(\nu(s)) \in \mathsf{inst}_I$. Since $S \sqsubseteq^*_\mathcal{R} R$, $\Xi_\mathcal{K}$ contains rules (5.3), and I is closed under $\Xi_\mathcal{K}$, we have that $\mathbb{S}_R(\nu(s)) \in \mathsf{inst}_I$. Because $\iota(\nu(s)) = \tau(s)$, by Lemma 5.12 we have that $\mathbb{S}_R(\tau(s)) \in \mathsf{inst}_J$ and $R(s,t)$ is a good atom, which contradicts our assumption.

In line 8, let P be an arbitrary role such that $\rho \in \mathcal{L}(P)$, $P \sqsubseteq^*_\mathcal{R} R$, and $m > 1$ implies that P is transitive. Since $\mathcal{K}$ is consistent, $\mathcal{R}$ is a role hierarchy, and $\top_r \not\sqsubseteq^*_\mathcal{R} R$, such a role P exists and $P \notin \{\top_r, \bot_r\}$. Furthermore, for each role T occurring in ρ, we have that $T \sqsubseteq^*_\mathcal{R} P$. Then, since $\rho(\nu(s), \nu(t)) \in \mathsf{inst}_I$ and I is closed under $\Xi_\mathcal{K}$, we also have that $P(\nu(s), \nu(t)) \in \mathsf{inst}_I$. Because $\iota(\nu(s)) = \tau(s)$ and $\iota(\nu(t)) = \tau(t)$, by Lemma 5.12, we then have $P(\tau(s), \tau(t)) \in \mathsf{inst}_J$; thus the conditions in line 8 are satisfied.

Next, consider the case in which $\langle s, t\rangle \notin \mathcal{E}$. Then either a vertex $v'' \in \mathcal{V}$ exists such that $\langle \nu(s), \nu(v'')\rangle \in \mathsf{dir}_I^+$ and $\langle \nu(v''), \nu(t)\rangle \in \mathsf{dir}_I^+$, or $\langle \nu(s), \nu(t)\rangle \notin \mathsf{dir}_I^+$. In either cases, we have that $m > 1$, and so P is transitive; hence the condition in line 9 is not satisfied.

Next, let $v_0 = s$, if s reaches t in $\mathcal{S}$; otherwise, let v_0 be the root that reaches t in $\mathcal{S}$. Furthermore, let ℓ_0 be the largest index in $[0..m]$ such that $w_{\ell_0} = \nu(v_0)$ and let $v_0, \dots, v_n$ be the unique path connecting v_0 to t in $\mathcal{S}$. By the form of the terms $w_{\ell_0+1}, \dots, w_m$, for each $j \in [\ell_0 + 1..m]$, we have that $\mathbb{D}_{R_j}(w_{j-1}, w_j) \in \mathsf{inst}_I$, as well as $\nu(v_0) = w_{\ell_0}$ and $\nu(v_n) = w_m$. Furthermore, for each $i \in [1..n]$, a unique index $\ell_i \in [\ell_0 + 1..m]$ exists such that $\nu(v_i) = w_{\ell_i}$. Now, let $\rho_0 = \chi_0 \cdots R_{\ell_0} \cdot \chi_{\ell_0}$ and for each $i \in [1..n]$, let $\rho_i = R_{\ell_{i-1}+1} \cdots R_{\ell_i}$. By property (h2) of Lemma 5.12, for each $j \in [\ell_{i-1} + 1..\ell_i]$ we have that $\mathbb{D}_{R_j}(\iota(w_{j-1}), \iota(w_j)) \in \mathsf{inst}_J$. Since $R_j \sqsubseteq^*_{\mathcal{R}} P$, $\mathsf{D}_{\mathcal{K}}$ contains rules (5.4), and J is closed under $\mathsf{D}_{\mathcal{K}}$, we also have that $\mathbb{D}_P(\iota(w_{j-1}), \iota(w_j)) \in \mathsf{inst}_J$. Finally, if P is not transitive, then $m = 1$. Therefore, role P is compatible with $\langle v_{i-1}, v_i\rangle \in \mathcal{E}$ and property ($\diamondsuit$) is satisfied in line 15.

We are left to show that property ($\heartsuit$) is satisfied in line 14. Then, assume that s does not reach t in $\mathcal{S}$, thus P is transitive, v_0 is the root that reaches t in $\mathcal{S}$, and $\nu(v_0) \in \mathsf{r\text{-}ind}_{\mathsf{D}_{\mathcal{K}}}$. Since $\rho \in \mathcal{L}(P)$, we also have $\rho_0 \in \mathcal{L}(P)$. By the form of the terms $w_0, \dots, w_{\ell_0}$, we have that $\nu(s) = w_0$ and $\nu(v_0) = w_{\ell_0}$; furthermore, for each $i \in [1..\ell_0]$, we have $R_i(w_{i-1}, w_i) \in \mathsf{inst}_I$; moreover, for each $i \in [0..\ell_0 - 1]$ and each role S occurring in χ_i, we have $S(w_i, w_i) \in \mathsf{inst}_I$. By property (h2) of Lemma 5.12 and by the construction of ν and τ, we have that $\rho_0(\tau(s), \nu(v_0)) \in \mathsf{inst}_J$. Since $\rho_0 \in \mathcal{L}(P)$ and J is closed under $\mathsf{D}_{\mathcal{K}}$, we also have $P(\tau(s), \nu(v_0)) \in \mathsf{inst}_J$ and property ($\heartsuit$) is satisfied. □

Chapter 7

Proof of Concept

To gain insight into the feasibility of our method for CQ answering, we implemented our algorithm for answering CQs over $\mathcal{ELHO}_{\bot}^{+}$ KBs in a prototypical system called EOLO. We believe that the practicability of our approach is mainly determined by three factors:

- the number of consequences captured by the datalog program $\mathsf{D}_{\mathcal{K}}$,
- the number of unsound answers obtained by evaluating CQs over $\mathsf{D}_{\mathcal{K}}$, and
- the time and the nondeterminism required by our filtering procedure.

Our experiments show that the overhead caused by materialising the consequences of $\mathsf{D}_{\mathcal{K}}$ is reasonably small in typical cases. Furthermore, even without transitive and reflexive roles, some queries can be challenging for our system. These queries, however, have been artificially designed to stress test combined approaches by generating large numbers of unsound answers, so it is not clear whether these queries are representative of information requests in typical applications. Nevertheless, our system can efficiently deal with the vast majority of the queries that we have tested, thus suggesting that our algorithm provides a practical basis for answering conjunctive queries in an expressive fragment of OWL 2 EL.

7.1 Test Setting and Benchmarks

We implemented EOLO in Java and ran our experiments on a Mid 2010 MacBook Pro with 4GB of RAM and an Intel Core 2 Duo 2.4 GHz processor. Our prototype uses the

Table 7.1: Statistics on the knowledge bases used in the experiments

	Concepts	Roles (transitive)	Nominals	TBox axioms	RBox axioms	Individuals	Unary atoms	Binary atoms
L5	132	32 (1)	9	235	7	100,848	169,079	296,941
L10						202,387	339,746	598,695
L20						426,144	714,692	1,259,936
SEM	60	16 (0)	4	209	0	17,945	47,248	65,193

RDFox [79, 80] system to materialise all consequences of the datalog program $\mathsf{D}_{\mathcal{K}}$. We used two different benchmarks for our experiments, which we present next.

We used a slightly modified version of the LSTW benchmark [73] to test our system with knowledge bases and queries that contain transitive roles. The LSTW benchmark consists of an OWL 2 QL version of the LUBM terminology [44], queries $q_1^l, \ldots, q_{11}^l$, and a data generator. The LSTW TBox extends the standard LUBM terminology with several axioms of type (4) (see Table 5.1 on page 53). To obtain an $\mathcal{ELHO}_\bot^+$ TBox, we first removed inverse roles and datatypes, thus obtaining an $\mathcal{ELH}_\bot$ terminology. Then, we added one axiom of type (2) (see Table 5.1), and 11 axioms using nine freshly introduced nominals. These additional axioms were designed to test the overhead caused by equality in the datalog program. Finally, we extended the resulting TBox by making the role *subOganizationOf* transitive, and by adding an axiom of type (4) and (10) over this role, thus obtaining an $\mathcal{ELHO}_\bot^+$ terminology. We used the data generator provided by LSTW to generate KBs L5, L10, and L20 of 5, 10, and 20 universities, respectively. The upper part of Table 7.1 shows statistics about these KBs. From the 11 LSTW queries, we did not consider queries q_4^l, q_6^l, and q_{11}^l because their result sets were empty: q_4^l relies on existential quantification over inverse roles, and the other two are empty already w.r.t. the original LSTW terminology. Query q_2^l contains a 'fork' and was designed to produce only unsound answers and thus stress the system. Finally, only queries q_3^l and q_7^l from the LSTW benchmark use transitive roles, so we have manually created four additional queries $q_{12}^l, \ldots, q_{15}^l$ which are shown in the upper part of Figure 7.1.

We also used a variant of the SEMINTEC benchmark to evaluate the performance of our system over knowledge bases that contain neither transitive nor reflexive roles, but require

$$
\begin{aligned}
q^l_{12} &= \exists \vec{y}.\, \mathsf{Professor}(x_1) \wedge \mathsf{teacherOf}(x_1, y_1) \wedge \mathsf{GraduateCourse}(y_1) \\
&\quad \wedge \mathsf{takesCourse}(x_2, y_1) \wedge \mathsf{memberOf}(x_2, y_2) \wedge \mathsf{subOrganizationOf}(y_2, y_3) \\
q^l_{13} &= \mathsf{ResearchGroup}(x_1) \wedge \mathsf{subOrganizationOf}(x_1, x_2) \wedge \mathsf{Organization}(x_2) \\
q^l_{14} &= \exists y.\, \mathsf{ResearchGroup}(x_1) \wedge \mathsf{subOrganizationOf}(x_1, y) \wedge \mathsf{University}(y) \\
&\quad \wedge \mathsf{subOrganizationOf}(x_2, y) \wedge \mathsf{Department}(x_2) \wedge \mathsf{affiliatedOrganizationOf}(x_1, y) \\
q^l_{15} &= \exists \vec{y}.\, \mathsf{subOrganizationOf}(y_1, y_2) \wedge \mathsf{memberOf}(x, y_1) \wedge \mathsf{Professor}(x) \\
&\quad \wedge \mathsf{subOrganizationOf}(y_3, y_2) \wedge \mathsf{memberOf}(x, y_3)
\end{aligned}
$$

$$
\begin{aligned}
q^s_6 &= \exists \vec{y}.\, \mathsf{hasLoan}(x_1, y_1) \wedge \mathsf{hasLoan}(x_2, y_2) \wedge \mathsf{Loan}(y_1) \wedge \mathsf{Loan}(y_2) \\
&\quad \wedge \mathsf{hasLoanStatusValue}(y_1, y_3) \wedge \mathsf{hasLoanStatusValue}(y_2, y_3) \\
q^s_7 &= \exists y.\, \mathsf{isOwnerOf}(x_1, x_2) \wedge \mathsf{Account}(x_2) \wedge \mathsf{hasSexValue}(x_1, y) \wedge \mathsf{MaleSex}(y) \\
q^s_8 &= \exists y.\, \mathsf{Account}(x) \wedge \mathsf{hasOwner}(x, y_1) \wedge \mathsf{hasAgeValue}(y_1, y_2)
\end{aligned}
$$

Figure 7.1: Our conjunctive queries q^l_{12}–q^l_{15} and q^s_6–q^s_9

equality reasoning due to nominals. The SEMINTEC benchmark consists of an $\mathcal{ALCOIF}$ knowledge base about financial services and acyclic queries q^s_1–q^s_5 developed within the SEMINTEC project at the University of Poznan.[1] The SEMINTEC ABox consists of approximately 65,000 assertions concerning 18,000 individuals. We obtained an $\mathcal{ELHO}_\bot$ knowledge base by first removing inverse roles, role functionality axioms, and universal restrictions from the original terminology, and then by adding nine axioms of type (4) (see Table 5.1), and six axioms using four freshly introduced nominals. The lower part of Table 7.1 shows structural information about the resulting $\mathcal{ELHO}_\bot$ knowledge base. Moreover, we manually crafted three additional queries q^s_6–q^s_8 that are shown in the lower part of Figure 7.1. Queries q^s_6–q^s_8 were designed to retrieve large number of candidate answers, thus stressing our filtering procedure. In particular, query q^s_6 resembles query q^l_2 from LSTW.

The test data, the queries, and the prototype system are all available online.[2]

7.2 Testing the Size of Materialisations

One of the aspects that, in our opinion, determines the practicability of our approach is the number of consequences of $\mathsf{D}_\mathcal{K}$, which should not be 'too large'. Table 7.2 shows

[1] http://www.cs.put.poznan.pl/alawrynowicz/semintec.htm
[2] http://www.cs.ox.ac.uk/isg/tools/EOLO/

Table 7.2: Size of the materialisations

	Individuals (% in $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$)	Unary atoms (% over $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$)	Binary atoms (% in $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$)	Total atoms (Ratio)
L5 before	100,848	169,079	296,941	466,020
after	100,873 (0.02)	511,115 (0.02)	1,343,848 (68.30)	1,854,963 (3.98)
L10 before	202,387	339,746	598,695	938,441
after	202,412 (0.01)	1,026,001 (0.01)	2,714,214 (68.37)	3,740,215 (3.98)
L20 before	426,144	714,692	1,259,936	1,974,628
after	426,169 (0.01)	2,157,172 (0.01)	5,720,670 (68.40)	7,877,842 (3.99)
SEM before	17,945	17,945	47,248	65,193
after	17,953 (0.1)	61,510 (0.04)	153,227 (57.45)	214,737 (3.29)

the materialisation results for our two benchmarks: the first column shows the number of individuals before and after materialisation and the percentage of individuals in $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$, the second column shows the number of unary atoms before and after materialisation and the percentage of atoms involving individuals in $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$, the third column does the same for binary atoms, and the fourth column shows the total number of atoms before and after materialisation and the ratio between the two. As one can see, for each input knowledge base, the materialisation step introduces few auxiliary individuals and the total number of atoms grows at most by a factor of four, which we believe is acceptable in most practical scenarios. The number of unary facts involving an auxiliary individual does not change with the size of the ABox, whereas the number of such binary facts increases by a constant factor. This is because, in axioms of type (4), atoms $A(o_{R,A})$ do not contain a variable, whereas atoms $R(x, o_{R,A})$ and $\mathbb{D}_R(x, o_{R,A})$ do. Finally, materialisation can be done in about 10 seconds for the largest knowledge base L20 of the LSTW benchmark, and in less than 1 second for the SEMINTEC knowledge base.

7.3 Testing the Filtering Procedure

Another important aspect that determines the feasibility of our CQ answering method is the 'practical hardness' of our filtering procedure, which can be evaluated by measuring the number of unsound answers obtained by evaluating CQs over $\mathsf{D}_\mathcal{K}$, and the time and the nondeterminism required by our filtering procedure. Tables 7.3a, 7.3b, and 7.3c show the

Table 7.3: The evaluation results for the LSTW and SEMINTEC benchmark, where column (C) shows the total number of candidate answers, column (U) shows the percentage of unsound candidate answers, column (F) shows the average time in ms required to filter each candidate answer, and column (N) shows the average number of nondeterministic choices required for each candidate answer.

(a) LSTW results for queries that do not use transitive roles

	q_1^l				q_2^l				q_5^l				q_8^l				q_9^l				q_{10}^t			
	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N
L5	111.9K	4.0	0.009	0	3.6M	100	0.010	0	27.9K	0	0.003	0	9.6K	0	0.002	0	1.1K	0	0.003	0	3.2K	0	0.001	0
L10	223.5K	4.2	0.009	0	32.0M	100	0.009	0	57.4K	0	0.002	0	19.4K	0	0.002	0	2.2K	0	0.005	0	6.4K	0	0.001	0
L20	487.3K	4.3	0.006	0	170.3M	100	0.009	0	121.2K	0	0.002	0	41.2K	0	0.002	0	4.8K	0	0.007	0	13.7K	0	0.001	0

(b) LSTW results for queries that use transitive roles

	q_3^l				q_7^l				q_{12}^l				q_{13}^l				q_{14}^l				q_{15}^l			
	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N
L5	10	0	0.001	0	19K	0	2.845	5.8	73K	12	1.71	7.55	3K	0	0.01	0	157K	66	1.07	8.6	30K	63	2.44	10.9
L10	22	0	0.001	0	38K	0	2.808	5.8	149K	12	1.68	7.54	6K	0	0.01	0	603K	81	1.20	9.6	61K	63	2.44	10.9
L20	43	0	0.001	0	82K	0	2.800	5.8	313K	12	1.66	7.55	12K	0	0.01	0	2.6M	90	1.28	10.3	129K	63	2.44	10.9

(c) SEMINTEC results

	q_1^s				q_2^s				q_3^s				q_4^s				q_5^s				q_6^s				q_7^s				q_8^s			
	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N
SEM	7	0	0.001	0.0	53	0	0.01	0	16	0	0.125	0	12	0	0.001	0	31	0	0.096	0	838K	55	0.004	0	2.2K	0	0.006	0	13K	0	0.004	0

results of our experiments for the queries in the LSTW and SEMINTEC benchmarks. Each table provides the following information for each query and each knowledge base.

- Column (C) shows the total number of candidate answers.
- Column (U) shows the percentage of candidate answers that are unsound.
- Column (F) shows the average time in ms required to filter candidate answers—that is, the total time required to filter all the candidate answers divided by the total number of candidate answers.
- Column (N) shows the average number of nondeterministic choices required to check the soundness of candidate answers—that is, we tally the total number of choices for the independent guesses in Algorithm 1 required to filter all the candidate answers, and then we divide this total number by the number of candidate answers.

As one can see from Table 7.3a, among the queries that do not use transitive roles in the LSTW benchmark, only query q_2^l retrieves a significant percentage of unsound candidate answers. This query has proven to be challenging for our system due to the large number of candidate answers, with an evaluation time of about 50 minutes over the largest knowledge base (L20). All other queries were evaluated over the largest knowledge base in less than 12 seconds. Furthermore, testing soundness of a candidate answer can typically be done in as few as several microseconds, and the test does not involve nondeterministic guessing.

Among the queries that use transitive roles in the LSTW benchmark, Table 7.3b shows that queries q_{14}^l and q_{15}^l retrieve a significant percentage of unsound candidate answers. As one can see from Figure 7.1 both these queries contain 'forks', hence the large rate of unsound answers. With an evaluation time of about 52 minutes over L20, only query q_{14}^l challenged our system due to the percentage of unsound candidate answers increasing with the size of the ABox and the overall large number of candidate answers to filter. For all the other queries, the percentage of unsound candidate answers does not increase with the the size of the ABox and the queries can be evaluated in less than 8 minutes over L20. In all cases, the soundness of a candidate answer can typically be tested in as few as several milliseconds, and the test involves a manageable number of nondeterministic choices.

For the SEMINTEC benchmark, Table 7.3c shows that only query q_6^s retrieve a significant percentage of unsound candidate answers. This query, however, did not challenge our system with an evaluation time of about 8 seconds. All other queries can be evaluated in less than 0.5 seconds. Moreover, similarly as for the queries without transitive roles in LSTW, testing soundness of a candidate answer can typically be done in few microseconds, and the test does not involve nondeterministic guessing.

Therefore, while some queries may be challenging, our experiments suggest that our CQ answering method can be practicable in many cases. Moreover, we believe that the performance of our system for queries that retrieve large numbers of unsound candidate answers can be further improved. In our experiments, we observed that, for a given query, the candidate answers that are unsound tend to share common regular patterns. So, while answering queries, one could cache unsound candidate answers, and then check whether a given candidate answer matches the pattern of a cached unsound answers, thus avoiding filtering similar candidate answers multiple times.

Chapter 8

Encoding Regular RBoxes Succinctly Using Bounded-Stack PDAs

Each reasoning algorithm for a DL with regular role inclusions known to us uses a step that checks whether $\rho \in \mathcal{L}(R)$ holds for an arbitrary role chain ρ and a role R. For example, to check whether $\mathcal{K} \models_{\mathcal{DL}} R(a, b)$ holds, an algorithm must ensure that, in each model of $\mathcal{K}$, a role chain $\rho \in \mathcal{L}(R)$ exists connecting the elements interpreting a and b. Although role inclusions characterise languages $\mathcal{L}(R)$, they do not lend themselves well to language recognition, so all algorithms known to us transform role inclusions into another, more manageable form. This is analogous to the fact that, while regular expressions characterise regular languages, the former are routinely transformed into FAs in order to facilitate language recognition.

Horrocks and Sattler [51] showed that, for each regular RBox $\mathcal{R}$ and each role R occurring in $\mathcal{R}$, one can construct an FA $\mathcal{F}_R$ such that $\mathcal{L}(\mathcal{F}_R) = \mathcal{L}(R)$. These FAs are used in a tableau decision procedure for $\mathcal{SROIQ}$. Given a $\mathcal{SROIQ}$ knowledge base, the tableau procedure tries to construct a finite graph representing a model of the KB, in which edges are labelled by roles, and vertices are labelled by concepts. The aforementioned FAs are used to ensure that the universal restrictions $\forall R.C$ labelling nodes obey the constraints im-

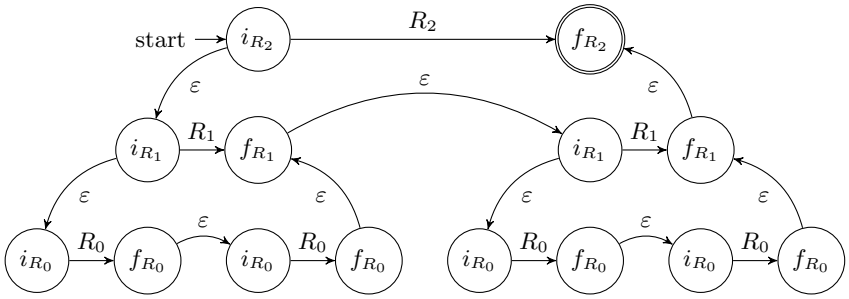

Figure 8.1: The FA $\mathcal{F}_{R_2}$ as constructed following Horrocks and Sattler [51]

posed by role inclusions; roughly speaking, this is obtained by running $\mathcal{F}_R$ over the graph produced by the tableaux algorithm while updating the current state of $\mathcal{F}_R$ along the path, and by labelling each reachable vertex in which the state of $\mathcal{F}_R$ is final with concept C. Simančík [92] optimised the tableau procedure by simulating FAs on-the-fly, rather than precomputing them in advance.

Horrocks and Sattler [51] observed that their FAs can contain exponentially many states. Kazakov [54] proved that this is unavoidable in some cases: for the regular RBox $\mathcal{R}_n$ containing axioms (8.1), the size of each FA $\mathcal{F}$ with $\mathcal{L}(\mathcal{F}) = \mathcal{L}(R_n)$ is exponential in n.

$$R_{i-1} \cdot R_{i-1} \sqsubseteq R_i \qquad \forall i \in [1..n] \tag{8.1}$$

This blowup in the number of states is caused by the simple model of computation underlying FAs, where the behaviour of the automaton is determined solely by the current state. In the example above, we have $\rho \in \mathcal{L}(R_n)$ whenever ρ consists of R_i repeated j times for some $i \in [0..n]$ with $j = 2^{n-i}$. Thus, while recognising such ρ, the FA recognising $\mathcal{L}(R_n)$ must 'remember' the number of occurrences of R_i it has already seen, which can be achieved only by using a different state for each number between 0 and 2^n. Figure 8.1 shows the FA $\mathcal{F}_{R_2}$ constructed by Horrocks and Sattler [51]: to 'remember' the current state, $\mathcal{F}_{R_2}$ contains two copies of automaton $\mathcal{F}_{R_1}$, and each copy of $\mathcal{F}_{R_1}$ contains two copies of automaton $\mathcal{F}_{R_0}$.

This blowup means that all the existing procedures for answering conjunctive queries over $\mathcal{ELRO}^+_\bot$ knowledge bases run in ExpTime in the worst-case. Our goal, however, is to show that the problem is PSpace-complete, and so we must devise a more succinct representation for the languages induced by role inclusions. Towards this goal, we note that

role inclusions are closely related to context-free grammars, and that context-free languages can be efficiently recognised using pushdown automata [50]—that is, FAs extended with an infinite stack for storing contextual information. Thus, given a regular RBox $\mathcal{R}$ and a role R occurring in $\mathcal{R}$, we construct a PDA $\mathcal{P}_R$ that accepts $\mathcal{L}(R)$. Unlike the finite automaton shown in Figure 8.1 that 'remembers' contextual information using states, PDA $\mathcal{P}_R$ uses the stack to 'remember' the current status of the computation and determine how to proceed. We show that the number of states in $\mathcal{P}_R$ is polynomial in the size of $\mathcal{R}$, and that $\mathcal{P}_R$ can recognise $\mathcal{L}(R)$ by using a stack of size linear in the size of $\mathcal{R}$; thus, $\mathcal{P}_R$ provides us with the required succinct encoding of $\mathcal{F}_R$.

8.1 Formalisation

We next formalise the intuitions that we have just presented and show how to associate each role R occurring in a regular RBox $\mathcal{R}$ with a PDA $\mathcal{P}_R$ that succinctly encodes $\mathcal{L}(R)$. In our construction of $\mathcal{P}_R$, only the initial and final states and stacks of $\mathcal{P}_R$ depend on the role R, while the derivation relation and all other components of $\mathcal{P}_R$ depend solely on the RBox $\mathcal{R}$. In the rest of this chapter, we fix an arbitrary regular RBox $\mathcal{R}$ such that, for each role inclusion $\rho \sqsubseteq R$ in $\mathcal{R}$, we have that $|\rho| \leq 2$, $\rho \neq R$, and $R \neq \top_r$. For each role $R \in \mathsf{rol}_{\mathcal{R}}$, we next define the PDA $\mathcal{P}_R$.

Definition 8.1. *For $R \in \mathsf{rol}_{\mathcal{R}}$ a role, $\mathcal{P}_R = \langle Q_{\mathcal{R}}, \mathsf{rol}_{\mathcal{R}}, \Gamma_{\mathcal{R}}, \delta_{\mathcal{R}}, i_R, \bot, f_R, \bot \rangle$ is the PDA where $Q_{\mathcal{R}} = \{i_T, f_T \mid T \in \mathsf{rol}_{\mathcal{R}}\}$ is the set of states, $\Gamma_{\mathcal{R}} = Q_{\mathcal{R}} \cup \{\bot\}$ is the stack alphabet, and $\delta_{\mathcal{R}}$ is the smallest transition function that satisfies the following conditions for each stack symbol $X \in \Gamma_{\mathcal{R}}$.*

(r) For each $T \in \mathsf{rol}_{\mathcal{R}} \setminus \{\top_r\}$, we have $\langle f_T, X \rangle \in \delta_{\mathcal{R}}(i_T, T, X)$.

(t1) For each $\epsilon \sqsubseteq T \in \mathcal{R}$, we have $\langle f_T, X \rangle \in \delta_{\mathcal{R}}(i_T, \varepsilon, X)$.

(t2) For each $T \cdot T \sqsubseteq T \in \mathcal{R}$, we have $\langle i_T, X \rangle \in \delta_{\mathcal{R}}(f_T, \varepsilon, X)$.

(t3) For each $T_1 \cdot T \sqsubseteq T \in \mathcal{R}$, we have $\langle i_{T_1}, i_T \cdot X \rangle \in \delta_{\mathcal{R}}(i_T, \varepsilon, X)$.

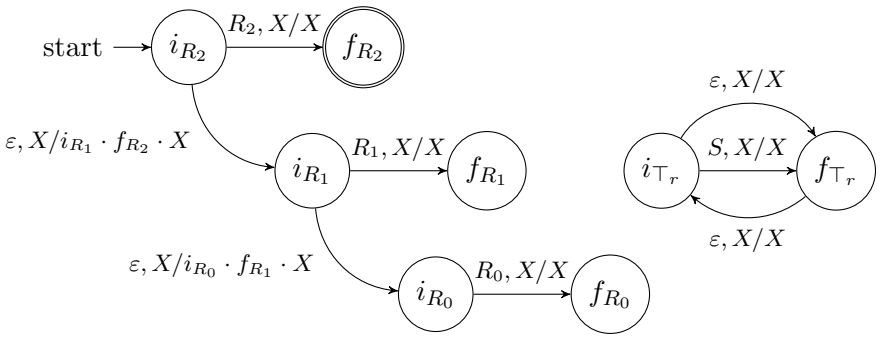

Figure 8.2: The PDA $\mathcal{P}_{R_2}$ corresponding to $\mathcal{F}_{R_2}$ where $X \in \Gamma_{\mathcal{R}}$ and $S \in \mathsf{rol}_{\mathcal{R}}$

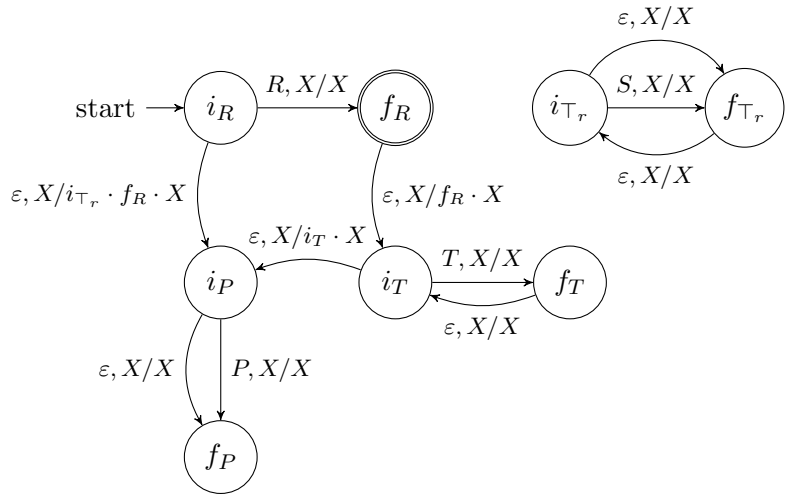

Figure 8.3: The PDA $\mathcal{P}_R$ for the RBox in Example 8.2 where $X \in \Gamma_{\mathcal{R}}$ and $S \in \mathsf{rol}_{\mathcal{R}}$

(t4) *For each* $T_1 \sqsubseteq T \in \mathcal{R}$, *we have* $\langle i_{T_1}, f_T \cdot X\rangle \in \delta_{\mathcal{R}}(i_T, \varepsilon, X)$, *and for each* $T_1 \cdot T_2 \sqsubseteq T \in \mathcal{R}$, *we have* $\langle i_{T_1}, i_{T_2} \cdot f_T \cdot X\rangle \in \delta_{\mathcal{R}}(i_T, \varepsilon, X)$.

(t5) *For each* $T \cdot T_2 \sqsubseteq T \in \mathcal{R}$, *we have* $\langle i_{T_2}, f_T \cdot X\rangle \in \delta_{\mathcal{R}}(f_T, \varepsilon, X)$.

(ur) *For each* $T \in \mathsf{rol}_{\mathcal{R}}$, *we have* $\langle f_{\top_r}, X\rangle \in \delta_{\mathcal{R}}(i_{\top_r}, T, X)$.

(u1) $\langle f_{\top_r}, X\rangle \in \delta_{\mathcal{R}}(i_{\top_r}, \varepsilon, X)$.

(u2) $\langle i_{\top_r}, X\rangle \in \delta_{\mathcal{R}}(f_{\top_r}, \varepsilon, X)$.

(p) *For each* $T \in \mathsf{rol}_{\mathcal{R}}$ *and each* $\lambda \in Q_{\mathcal{R}}$, *we have* $\langle \lambda, \epsilon\rangle \in \delta_{\mathcal{R}}(f_T, \varepsilon, \lambda)$.

In the following examples, we present the PDA that succinctly encodes the FA $\mathcal{F}_{R_2}$ shown in Figure 8.1, and we explain the different types of transitions in Definition 8.1, and how the content of the stack influences the computation of PDAs.

Example 8.1. *Consider the RBox* $\mathcal{R}_2 = \{R_0 \cdot R_0 \sqsubseteq R_1,\ R_1 \cdot R_1 \sqsubseteq R_2\}$. *Figure 8.1 shows the FA* $\mathcal{F}_{R_2}$ *that recognises* $\mathcal{L}(R_2)$, *whereas Figure 8.2 shows the PDA* $\mathcal{P}_{R_2}$ *corresponding to* $\mathcal{F}_{R_2}$. *A transition* $\langle \lambda', \gamma \rangle \in \delta_{\mathcal{R}}(\lambda, c, X)$ *is shown as* $\lambda \xrightarrow{c, X/\gamma} \lambda'$, *where* X/γ *indicates that the transition replaces the top-most stack symbol* X *with word* γ; *moreover, transitions of the form (p) from Definition 8.1 are not shown in the figure for the sake of clarity. As one can see from the figure, unlike in FA* $\mathcal{F}_{R_2}$, *there is no copying of states in PDA* $\mathcal{P}_{R_2}$.

Example 8.2. *To explain the different types of transitions in Definition 8.1 and how the stack is used in the computation of a PDA, we use the regular RBox* $\mathcal{R}$ *containing role inclusions (8.2)–(8.6). Figure 8.3 shows PDA* $\mathcal{P}_R$ *using the notation from Example 8.1.*

$$\epsilon \sqsubseteq P \tag{8.2}$$

$$T \cdot T \sqsubseteq T \tag{8.3}$$

$$P \cdot \top_r \sqsubseteq R \tag{8.4}$$

$$R \cdot T \sqsubseteq R \tag{8.5}$$

$$P \cdot T \sqsubseteq T \tag{8.6}$$

Each role T *is associated with states* i_T *and* f_T, *and moving from the former to the latter ensures that the PDA reads a role chain* $\rho \in \mathcal{L}(T)$. *A transition of type (r) allows the PDA to read* T *in state* i_T. *An* ε*-transition of type (t1) from* i_T *to* f_T *is added if* T *is reflexive, and it allows the PDA to read the empty role chain; in our example, axiom (8.2) introduces the* ε*-transition from* i_P *to* f_P. *Moreover, an* ε*-transition of type (t2) from* f_T *to* i_T *is added if* T *is transitive, and it allows the PDA to read any sequence of role chains* $\rho_1, \dots, \rho_n \in \mathcal{L}(T)$; *in our example, axiom (8.3) introduces the* ε*-transition from* f_T *to* i_T. *Transitions of types (ur), (u1), and (u2) analogously reflect the properties of* $\top_r$: *(ur) allows the PDA to read an arbitrary role, and (u1) and (u2) reflect the reflexivity and transitivity of* $\top_r$, *respectively. None of these transitions affect the PDA's stack.*

To illustrate transitions of type (t4), we next show how, for $\rho_1 = P \cdot R$, *PDA* $\mathcal{P}_R$ *determines that* $\rho_1 \in \mathcal{L}(R)$; *the latter is ensured by axiom (8.4). Assume that PDA* $\mathcal{P}_R$ *is in state* i_R *with* $\bot$ *on its stack. Due to axiom (8.4),* $\mathcal{P}_R$ *can make an* ε*-transition of type*

(t4) to state i_P, pushing $i_{\top_r} \cdot f_R$ on the stack. Since the new state is i_P, the PDA will next need to read P; furthermore, the stack content signals to the PDA that, after it finishes reading P, it should move to state $i_{\top_r}$ to read $\top_r$ and then to state f_R to finish reading R. Indeed, $\mathcal{P}_R$ can then make a transition of type (r) to state f_P to read P, followed by an ε-transition of type (p) to state $i_{\top_r}$ popping $i_{\top_r}$ off the stack; next, the PDA can make a transition of type (ur) to state $f_{\top_r}$ reading R, followed by an ε-transition of type (p) to state f_R popping f_R off the stack. At this point, the PDA accepts the input.

To illustrate transitions of types (t3) and (t5), we next show how, for $\rho_2 = R \cdot P \cdot T$, PDA $\mathcal{P}_R$ determines that $\rho_2 \in \mathcal{L}(R)$; the latter is ensured by axioms (8.5) and (8.6). Again, assume that PDA $\mathcal{P}_R$ is in state i_R with $\bot$ on its stack. PDA $\mathcal{P}_R$ can then make a transition of type (r) to state f_R, reading R and leaving the stack unchanged; next, due to axiom (8.5), $\mathcal{P}_R$ can make an ε-transition of type (t5) to state i_T, pushing f_R on the stack. Due to axiom (8.6), PDA $\mathcal{P}_R$ can next make an ε-transition of type (t3) to state i_P, pushing i_T on the stack; at this point, the stack contains $i_T \cdot f_R \cdot \bot$. Next, the PDA can make a transition of type (r) to state f_P reading P, and then an ε-transition of type (p) to state i_T popping i_T off the stack; furthermore, in an analogous way, the PDA can move to state f_T reading T and leaving $f_R \cdot \bot$ on the stack. Finally, the PDA can make an ε-transition of type (p) to state f_R popping f_R off the stack. At this point, the PDA accepts the input.

To understand the benefit of using PDAs rather than FAs, note that $\mathcal{P}_R$ reaches state i_P while recognising both ρ_1 and ρ_2. Role P occurs in axioms (8.4) and (8.6), so when $\mathcal{P}_R$ moves into state i_P in order to read an occurrence of P, it must 'remember' which of the two axioms caused the move so that it knows how to continue after reading P: for ρ_1, $\mathcal{P}_R$ must must continue reading $\top_r$, whereas for ρ_2, it must continue reading T. Unlike the FAs by Horrocks and Sattler [51] that remember this information by copying states, $\mathcal{P}_R$ remembers this information on its stack: for ρ_1, it reaches i_P with $i_{\top_r} \cdot f_R \cdot \bot$ on its stack, whereas for ρ_2, $\mathcal{P}_R$ reaches i_P with $i_T \cdot f_R \cdot \bot$ on its stack. Thus, the stack of $\mathcal{P}_R$ is analogous to stacks in programming languages: stack symbols correspond to return addresses, and transitions of type (p) correspond to 'return' statements.

The following result is immediate from the definition of PDA $\mathcal{P}_R$.

Proposition 8.2. *PDA $\mathcal{P}_R$ can be computed in time polynomial in $|\mathcal{R}|$.*

The following theorem states that PDA $\mathcal{P}_R$ accepts $\mathcal{L}(R)$ and that $\mathcal{P}_R$ has stack bounded by the size of $\mathcal{R}$. The proof of this result is given in Section 8.2.

Theorem 8.3. *For each role $R \in \mathsf{rol}_{\mathcal{R}}$ and each role chain ρ,*

1. *$\rho \in \mathcal{L}(\mathcal{P}_R)$ if and only if $\rho \in \mathcal{L}(R)$, and*
2. *$\mathcal{P}_R$ has stack bounded by $2 \cdot \mathsf{lv}(R) + 1$.*

Theorem 8.3 gives rise to the following notion of the depth of RBox $\mathcal{R}$, which provide us with a global bound on the stack size of the PDAs encoding $\mathcal{R}$.

Definition 8.4. *The* depth of the RBox $\mathcal{R}$ *is* $\mathsf{d}_{\mathcal{R}} = \max_{R \in \mathsf{rol}_{\mathcal{R}}}(2 \cdot \mathsf{lv}(R) + 1)$.

Finally, we outline how our bounded-stack encoding of regular RBoxes can reduce the space used by the tableau algorithm for $\mathcal{SROIQ}$. Since $\mathcal{ELRO}_{\bot}^{+}$ does not support inverse roles, Definition 8.1 does not directly provide us with an encoding of the languages induced by $\mathcal{SROIQ}$ RBoxes. Nevertheless, we can extend the construction above by 'completing' RBox $\mathcal{R}$ so that $\mathsf{inv}(R_n) \cdots \mathsf{inv}(R_1) \sqsubseteq \mathsf{inv}(R) \in \mathcal{R}$ for each role inclusion $R_1 \cdots R_n \sqsubseteq R$ in the RBox, where $\mathsf{inv}(\cdot)$ maps each role to its inverse. One can check that, for each (inverse) role R, the PDA $\mathcal{P}_R$ constructed using the completed RBox $\mathcal{R}$ encodes $\mathcal{F}_R$. Therefore, we can modify the portion of the tableau algorithm responsible for checking the satisfaction of universal restrictions by running a bounded-stack PDA over the graph constructed by the tableau procedure. Roughly speaking, for each universal restriction $\forall R.C$ labelling a vertex, we run $\mathcal{P}_R$ over the graph while updating the current state and the stack of $\mathcal{P}_R$, and we label each reachable vertex in which the current state and stack of $\mathcal{P}_R$ are final with concept C. Since $\mathcal{P}_R$ and its stack are of size polynomial in $|\mathcal{R}|$, this requires polynomial space, unlike the FAs by Horrocks and Sattler [51] and the optimised encoding by Simančík [92], which may require exponential space.

8.2 Proof of Correctness

In this section, we prove Theorem 8.3. To this end, let $\vdash$ be the derivation relation for the transition function $\delta_{\mathcal{R}}$; furthermore, for each derivation step $\langle \lambda, \rho, \gamma \rangle \vdash \langle \lambda', \rho', \gamma' \rangle$, we write

$\langle \lambda, \rho, \gamma \rangle \vdash_x \langle \lambda', \rho', \gamma' \rangle$ if $\langle \lambda', \rho', \gamma' \rangle$ can be obtained from $\langle \lambda, \rho, \gamma \rangle$ by applying a transition of the form (x) from Definition 8.1 with $x \in \{r, t1, \ldots, t5, ur, u1, u2, p\}$.

8.2.1 Soundness and Stack Boundedness

Next, we prove that, for each role $R \in \mathsf{rol}_{\mathcal{R}}$ and each role chain $\rho \in (\mathsf{rol}_{\mathcal{R}})^*$,

(1) $\rho \in \mathcal{L}(\mathcal{P}_R)$ implies that $\rho \in \mathcal{L}(R)$, and

(2) $\mathcal{P}_R$ has stack bounded by $2 \cdot \mathsf{lv}(R) + 1$.

To this end, we first show that PDA $\mathcal{P}_R$ satisfies the following *liveness property*: if during its computation $\mathcal{P}_R$ pushes a state $\lambda \in Q_{\mathcal{R}}$ on the stack, then $\mathcal{P}_R$ will eventually pop λ off the stack. Then, we show that each derivation of $\mathcal{P}_R$ moving from state i_R to state f_R takes one of five forms; we call such derivations *regular*. Finally, we show that regular derivations satisfy properties (1) and (2).

We start by showing that each PDA $\mathcal{P}_R$ satisfies the following liveness property.

Lemma 8.5. *Let $\langle \lambda_0, \rho_0, \gamma_0 \cdot \gamma \rangle \vdash \cdots \vdash \langle \lambda_n, \rho_n, \gamma_n \cdot \gamma \rangle$ be an arbitrary derivation such that $\lambda_0 = i_R$, $\lambda_n = f_R$, and $\gamma_0 = \epsilon$ for some role $R \in \mathsf{rol}_{\mathcal{R}}$ and some word $\gamma \in \Gamma_{\mathcal{R}}^*$. Then, for each role T such that $\mathsf{lv}(T) < \mathsf{lv}(R)$ and each $i \in [0..n]$ such that $\lambda_i \in \{i_T, f_T\}$ and $\gamma_i = \lambda_i' \cdot \gamma_i'$ with $\lambda_i' \in Q_{\mathcal{R}}$, an index $j \in [i..n]$ exists such that, for each $k \in [i..j]$, we have that*

(a) $\lambda_j = f_T$ and $\gamma_j = \gamma_i$;

(b) word γ_k is of the form $\gamma_k = \gamma_k'' \cdot \gamma_i$ for some $\gamma_k'' \in \Gamma_{\mathcal{R}}^*$; and,

(c) $\lambda_{j+1} = \lambda_i'$, $\gamma_{j+1} = \gamma_i'$, and $\rho_{j+1} = \rho_j$.

Proof. Let $\langle \lambda_0, \rho_0, \gamma_0 \cdot \gamma \rangle \vdash \cdots \vdash \langle \lambda_n, \rho_n, \gamma_n \cdot \gamma \rangle$ be as above, and for each $i \in [0..n-1]$, let $x_i \in \{r, t1, \ldots, t5, ur, u1, u2, p\}$ be the form of derivation step i—that is, we fix x_i (arbitrarily if there is more than one possibility) such that $\langle \lambda_i, \rho_i, \gamma_i \cdot \gamma \rangle \vdash_{x_i} \langle \lambda_{i+1}, \rho_{i+1}, \gamma_{i+1} \cdot \gamma \rangle$ holds. Furthermore, for each role T such that $\mathsf{lv}(T) < \mathsf{lv}(R)$, let I_T be the set containing each index $i \in [0..n]$ such that $\lambda_i \in \{i_T, f_T\}$ and γ_i is of the form $\gamma_i = \lambda_i' \cdot \gamma_i'$ with $\lambda_i' \in Q_{\mathcal{R}}$. Note that, for each index $i \in I_T$, due to $\mathsf{lv}(T) < \mathsf{lv}(R)$, $\lambda_i \in \{i_T, f_T\}$, and $\lambda_n = f_R$, we have that $i < n$—that is, $\langle \lambda_i, \rho_i, \gamma_i \cdot \gamma \rangle \vdash \langle \lambda_{i+1}, \rho_{i+1}, \gamma_{i+1} \cdot \gamma \rangle$ occurs in our derivation. Next, by

induction on $m \in \mathbb{N}$, we show that, for each role T with $m = \mathsf{lv}(T) < \mathsf{lv}(R)$ and each $i \in I_T$, some $j \in [i..n]$ exists satisfying properties (a)–(c).

Base case ($\diamondsuit$). Consider a role $T \in \mathsf{rol}_{\mathcal{R}}$ with $0 = \mathsf{lv}(T) < \mathsf{lv}(R)$. We consider the interesting case in which $I_T \neq \emptyset$; otherwise, properties (a)–(c) hold vacuously. Given that $\mathsf{lv}(T) = 0$ and $\lambda_i \in \{i_T, f_T\}$, we have that $x_i \in \{r, t1, t2, ur, u1, u2, p\}$. By reverse-induction on I_T (i.e., by induction starting from the maximal element), we next show that each index $i \in I_T$ satisfies the required properties.

Base case. Let $i = \max I_T$. Note that, if $x_i \in \{r, t1, t2, ur, u1, u2\}$, then $\lambda_{i+1} \in \{i_T, f_T\}$ and $\gamma_{i+1} = \gamma_i$; thus, we have that $i + 1 \in I_T$, which contradicts the maximality of i. The only remaining possibility is $x_i = p$, and so $\lambda_i = f_T$, $\lambda_{i+1} = \lambda'_i$, $\gamma_{i+1} = \gamma'_i$, and $\rho_{i+1} = \rho_i$; but then, $j = i$ satisfies properties (a)–(c).

Inductive step. Consider an arbitrary index $i \in I_T$ such that properties (a)–(c) hold for each $\ell \in I_T$ with $\ell > i$. If $x_i \in \{r, t1, t2, ur, u1, u2\}$, then $\lambda_{i+1} \in \{i_T, f_T\}$ and $\gamma_{i+1} = \gamma_i$; hence, $i_{i+1} \in I_T$ so, by the inductive hypothesis, an index j exists satisfying properties (a)–(c). Otherwise, if $x_i = p$, then $\lambda_i = f_T$, $\lambda_{i+1} = \lambda'_i$, $\gamma_{i+1} = \gamma'_i$, and $\rho_{i+1} = \rho_i$, so $j = i$ satisfies properties (a)–(c).

Inductive Step ($\diamondsuit$). Consider an arbitrary $m \in \mathbb{N}$ such that properties (a)–(c) hold for each role $P \in \mathsf{rol}_{\mathcal{R}}$ with $\mathsf{lv}(P) \leq m$ and $\mathsf{lv}(P) < \mathsf{lv}(R)$ and each index in I_P. Furthermore, consider an arbitrary role T such that $m + 1 = \mathsf{lv}(T) < \mathsf{lv}(R)$. We consider the interesting case where $I_T \neq \emptyset$; otherwise, properties (a)–(c) hold vacuously. Recall that for each $\rho \sqsubseteq R' \in \mathcal{R}$ we have $R' \neq \top_r$, so $\mathsf{lv}(\top_r) = 0$ and $T \neq \top_r$. Thus, each $i \in I_T$ is such that $x_i \notin \{ur, u1, u2\}$. By reverse-induction on I_T, we next show that each index $i \in I_T$ satisfies the required properties.

Base case ($\heartsuit$). Let $i = \max I_T$. If $x_i \in \{r, t1, t2\}$, then $\lambda_{i+1} \in \{i_T, f_T\}$ and $\gamma_{i+1} = \gamma_i$; thus, we have $i + 1 \in I_T$, which contradicts the maximality of i. If $x_i \in \{t3, t4, t5\}$, then $\lambda_{i+1} \in \{i_P, f_P\}$ for some role P such that $\mathsf{lv}(P) < \mathsf{lv}(T)$ and $\mathsf{lv}(P) < \mathsf{lv}(R)$; furthermore, we have that γ_{i+1} is of the form $\gamma_{i+1} = \gamma''_{i+1} \cdot \lambda_T \cdot \gamma_i$ where $\lambda_T \in \{i_T, f_T\}$ and γ''_{i+1} is a sequence of zero or one states. Each state γ occurring in γ''_{i+1} is such that $\gamma \in \{i_S, f_S\}$ for some role S of level less than T. But then, by the inductive hypothesis ($\diamondsuit$), an index $\ell > i$ exists such

that $\lambda_\ell = \lambda_T$ and $\gamma_\ell = \gamma_i$, which contradicts the maximality of i. Finally, if $x_i = p$, then $\lambda_i = f_T$, $\lambda_{i+1} = \lambda'_i$, $\gamma_{i+1} = \gamma'_i$, and $\rho_{i+1} = \rho_i$, so $j = i$ satisfies properties (a)–(c).

Inductive step ($\heartsuit$). Consider an arbitrary index $i \in I_T$ such that properties (a)–(c) hold for each index $\ell \in I_T$ with $\ell > i$, and consider the possible forms of x_i.

- $x_i \in \{r, t1, t2\}$. Then, $\lambda_{i+1} \in \{i_T, f_T\}$ and $\gamma_{i+1} = \gamma_i$, so $i+1 \in I_T$. By the inductive hypothesis ($\heartsuit$), an index j exists satisfying properties (a)–(c).

- $x_i = t3$. Then, $\lambda_{i+1} = i_{T_1}$ and $\gamma_{i+1} = i_T \cdot \gamma_i$ for some role T_1 with $\mathsf{lv}(T_1) < \mathsf{lv}(T)$. Thus, $i+1 \in I_{T_1}$. By the inductive hypothesis ($\diamondsuit$), an index $\ell \in [i+1..n]$ exists such that $\lambda_\ell = f_{T_1}$ and $\gamma_\ell = \gamma_{i+1}$; furthermore, for each $k \in [i+1..\ell]$, we have that γ_k is of the form $\gamma_k = \gamma''_k \cdot \gamma_{i+1}$ for some word $\gamma''_k \in \Gamma^*_{\mathcal{R}}$; finally, $\lambda_{\ell+1} = i_T$ and $\gamma_{\ell+1} = \gamma_i$. By the definition of I_T, we have that $\ell + 1 \in I_T$. By the inductive hypothesis ($\heartsuit$), an index j exists satisfying properties (a)–(c).

- $x_i = t4$. Then, $\lambda_{i+1} = i_{T_1}$ and one of the following two cases holds.

  - $\gamma_{i+1} = f_T \cdot \gamma_i$ and a role T_1 with $\mathsf{lv}(T_1) < \mathsf{lv}(T)$ exists such that $\lambda_{i+1} = i_{T_1}$. Thus, $i+1 \in I_{T_1}$. By the inductive hypothesis ($\diamondsuit$), an index $\ell \in [i+1..n]$ exists such that $\lambda_\ell = f_{T_1}$ and $\gamma_\ell = \gamma_{i+1}$; moreover, for each $k \in [i..\ell]$, we have that γ_k is of the form $\gamma_k = \gamma''_k \cdot \gamma_{i+1}$ for some word $\gamma''_k \in \Gamma^*_{\mathcal{R}}$; finally, $\lambda_{\ell+1} = f_T$ and $\gamma_{\ell+1} = \gamma_i$. By the definition of I_T, we have that $\ell + 1 \in I_T$. By the inductive hypothesis ($\heartsuit$), an index j exists satisfying properties (a)–(c).

  - $\gamma_{i+1} \neq f_T \cdot \gamma_i$ and role T_1 and T_2 with $\mathsf{lv}(T_1) < \mathsf{lv}(T)$ and $\mathsf{lv}(T_2) < \mathsf{lv}(T)$ exist such that $\lambda_{i+1} = i_{T_1}$ and $\gamma_{i+1} = i_{T_2} \cdot f_T \cdot \gamma_i$. Thus, $i+1 \in I_{T_1}$. By the inductive hypothesis ($\diamondsuit$), an index $\ell_1 \in [i+1..n]$ exists such that $\lambda_{\ell_1} = f_{T_1}$ and $\gamma_{\ell_1} = \gamma_{i+1}$; moreover, for each $k \in [i..\ell_1]$, we have that γ_k is of the form $\gamma_k = \gamma''_k \cdot \gamma_{i+1}$ for some word $\gamma''_k \in \Gamma^*_{\mathcal{R}}$; finally, $\lambda_{\ell_1+1} = i_{T_2}$ and $\gamma_{\ell_1+1} = f_T \cdot \gamma_i$. Consequently, $\ell_1 + 1 \in I_{T_2}$. Again, by the inductive hypothesis ($\diamondsuit$), an index $\ell_2 \in [\ell_1 + 1..n]$ exists such that $\lambda_{\ell_2} = f_{T_2}$ and $\gamma_{\ell_2} = \gamma_{\ell_1+1}$; furthermore, for each $k \in [\ell_1 + 1..\ell_2]$, we have that γ_k is of the form $\gamma_k = \gamma''_k \cdot \gamma_{\ell_1+1}$ for some word $\gamma''_k \in \Gamma^*_{\mathcal{R}}$; finally, $\lambda_{\ell_2+1} = f_T$ and $\gamma_{\ell_2+1} = \gamma_i$. By the definition of I_T, we have that $\ell_2 + 1 \in I_T$. By

the inductive hypothesis (♡), an index j exists satisfying properties (a)–(c).

- $x_i = t5$. Then, $\lambda_{i+1} = i_{T_2}$ and $\gamma_{i+1} = f_T \cdot \gamma_i$ for some role T_2 with $\mathsf{lv}(T_2) < \mathsf{lv}(T)$. Then, $i+1 \in I_{T_2}$. By the inductive hypothesis (♢), an index $\ell \in [i+1..n]$ exists such that $\lambda_\ell = f_{T_2}$ and $\gamma_\ell = \gamma_{i+1}$; for each $k \in [i..\ell]$, we have that γ_k is of the form $\gamma_k = \gamma''_k \cdot \gamma_{i+1}$ for some word $\gamma''_k \in \Gamma^*_{\mathcal{R}}$; finally, $\lambda_{\ell+1} = f_T$ and $\gamma_{\ell+1} = \gamma_i$. By the definition of I_T, we have that $\ell + 1 \in I_T$. So, by the inductive hypothesis (♡), an index j exists satisfying properties (a)–(c).

- $x_i = p$. Then, $\lambda_i = f_T$, $\lambda_{i+1} = \lambda'_i$, $\gamma_{i+1} = \gamma'_i$, and $\rho_{i+1} = \rho_i$. Therefore, $j = i$ satisfies properties (a)–(c). □

Next, for each role $R \in \mathsf{rol}_{\mathcal{R}}$, we define the notion of *regular derivations of* $\mathcal{P}_R$.

Definition 8.6. *The set of regular derivations of $\mathcal{P}_{\top_r}$ is inductively defined as follows, for each role $T \in \mathsf{rol}_{\mathcal{R}}$, each role chain $\rho_i \in (\mathsf{rol}_{\mathcal{R}})^*$, and each $\gamma \in \Gamma^*_{\mathcal{R}}$.*

s_{ur} *$\langle i_{\top_r}, T \cdot \rho_0, \gamma\rangle \vdash_{ur} \langle f_{\top_r}, \rho_0, \gamma\rangle$ is a regular derivation of $\mathcal{P}_{\top_r}$.*

s_{u1} *$\langle i_{\top_r}, \rho_0, \gamma\rangle \vdash_{u1} \langle f_{\top_r}, \rho_0, \gamma\rangle$ is a regular derivation of $\mathcal{P}_{\top_r}$.*

s_{u2} *If $\langle i_{\top_r}, \rho_0, \gamma\rangle \vdash \cdots \vdash \langle f_{\top_r}, \rho_k, \gamma\rangle$ and $\langle i_{\top_r}, \rho_k, \gamma\rangle \vdash \cdots \vdash \langle f_{\top_r}, \rho_n, \gamma\rangle$ are regular derivations of $\mathcal{P}_{\top_r}$, then the following is also a regular derivation of $\mathcal{P}_{\top_r}$.*

$$\langle i_{\top_r}, \rho_0, \gamma\rangle \vdash \cdots \vdash \langle f_{\top_r}, \rho_k, \gamma\rangle \vdash_{u2} \langle i_{\top_r}, \rho_k, \gamma\rangle \vdash \cdots \vdash \langle f_{\top_r}, \rho_n, \gamma\rangle$$

Next, consider an arbitrary natural number $m \in \mathbb{N}$ and assume that regular derivations of $\mathcal{P}_T$ have already been defined for $T = \top_r$ and each role $T \in \mathsf{rol}_{\mathcal{R}}$ such that $\mathsf{lv}(T) \leq m$. Then, for each role $R \in \mathsf{rol}_{\mathcal{R}}$ with $\mathsf{lv}(R) = m+1$, regular derivations of $\mathcal{P}_R$ *are defined as follows, for each $R_{(i)} \in \mathsf{rol}_{\mathcal{R}}$, each $\rho_i \in \mathsf{rol}^*_{\mathcal{R}}$, and each $\gamma \in \Gamma^*_{\mathcal{R}}$.*

s_r *$\langle i_R, R \cdot \rho_0, \gamma\rangle \vdash_r \langle f_R, \rho_0, \gamma\rangle$ is a regular derivation of $\mathcal{P}_R$.*

s_{t1} *If $\epsilon \sqsubseteq R \in \mathcal{R}$, then $\langle i_R, \rho_0, \gamma\rangle \vdash_{t1} \langle f_R, \rho_0, \gamma\rangle$ is a regular derivation of $\mathcal{P}_R$.*

s_{t2} *If* $R \cdot R \sqsubseteq R \in \mathcal{R}$ *and*

$$\langle i_R, \rho_0, \gamma\rangle \vdash \cdots \vdash \langle f_R, \rho_k, \gamma\rangle \text{ and } \langle i_R, \rho_k, \gamma\rangle \vdash \cdots \vdash \langle f_R, \rho_n, \gamma\rangle$$

are regular derivations of $\mathcal{P}_R$*, then the following is also a regular derivation of* $\mathcal{P}_R$*.*

$$\langle i_R, \rho_0, \gamma\rangle \vdash \cdots \vdash \langle f_R, \rho_k, \gamma\rangle \vdash_{t2} \langle i_R, \rho_k, \gamma\rangle \vdash \cdots \vdash \langle f_R, \rho_n, \gamma\rangle$$

s_{t3} *If* $R_1 \cdot R \sqsubseteq R \in \mathcal{R}$*,* $\langle i_{R_1}, \rho_0, i_R \cdot \gamma\rangle \vdash \cdots \vdash \langle f_{R_1}, \rho_k, i_R \cdot \gamma\rangle$ *is a regular derivation of* $\mathcal{P}_{R_1}$*, and* $\langle i_R, \rho_k, \gamma\rangle \vdash \cdots \vdash \langle f_R, \rho_n, \gamma\rangle$ *is a regular derivation of* $\mathcal{P}_R$*, then the following is also a regular derivation of* $\mathcal{P}_R$*.*

$$\begin{array}{lll}
\langle i_R, & \rho_0, & \gamma\rangle \quad \vdash_{t3} \\
\langle i_{R_1}, & \rho_0, & i_R \cdot \gamma\rangle \quad \vdash \cdots \vdash \quad \langle f_{R_1}, \rho_k, i_R \cdot \gamma\rangle \quad \vdash_p \\
\langle i_R, & \rho_k, & \gamma\rangle \quad \vdash \cdots \vdash \quad \langle f_R, \rho_n, \gamma\rangle
\end{array}$$

s_{t4} *If* $R_1 \sqsubseteq R \in \mathcal{R}$ *and* $\langle i_{R_1}, \rho_0, f_R \cdot \gamma\rangle \vdash \cdots \vdash \langle f_{R_1}, \rho_k, f_R \cdot \gamma\rangle$ *is a regular derivation of* $\mathcal{P}_{R_1}$*, then the following is also a regular derivation of* $\mathcal{P}_R$*.*

$$\begin{array}{lll}
\langle i_R, & \rho_0, & \gamma\rangle \quad \vdash_{t4} \\
\langle i_{R_1}, & \rho_0, & f_R \cdot \gamma\rangle \quad \vdash \cdots \vdash \quad \langle f_{R_1}, \rho_k, f_R \cdot \gamma\rangle \quad \vdash_p \\
\langle f_R, & \rho_k, & \gamma\rangle
\end{array}$$

If $R_1 \cdot R_2 \sqsubseteq R \in \mathcal{R}$*,* $\langle i_{R_1}, \rho_0, i_{R_2} \cdot f_R \cdot \gamma\rangle \vdash \cdots \vdash \langle f_{R_1}, \rho_k, i_{R_2} \cdot f_R \cdot \gamma\rangle$ *is a regular derivation of* $\mathcal{P}_{R_1}$*, and* $\langle i_{R_2}, \rho_k, f_R \cdot \gamma\rangle \vdash \cdots \vdash \langle f_{R_2}, \rho_n, f_R \cdot \gamma\rangle$ *is a regular derivation*

of $\mathcal{P}_{R_2}$, then the following is also a regular derivation of $\mathcal{P}_R$.

$$\begin{array}{llll}
\langle i_R,\ \rho_0, & \gamma\rangle & \vdash_{t4} & \\
\langle i_{R_1}, \rho_0, i_{R_2} \cdot f_R \cdot \gamma\rangle & \vdash \cdots \vdash & \langle f_{R_1}, \rho_k, i_{R_2} \cdot f_R \cdot \gamma\rangle & \vdash_p \\
\langle i_{R_2}, \rho_k, & f_R \cdot \gamma\rangle & \vdash \cdots \vdash \langle f_{R_2}, \rho_n, & f_R \cdot \gamma\rangle \quad \vdash_p \\
\langle f_R,\ \rho_n, & \gamma\rangle &&
\end{array}$$

s_{t5} *If $R \cdot R_2 \sqsubseteq R \in \mathcal{R}$, $\langle i_R, \rho_0, \gamma\rangle \vdash \cdots \vdash \langle f_R, \rho_k, \gamma\rangle$ is a regular derivation of $\mathcal{P}_R$, and $\langle i_{R_2}, \rho_k, f_R \cdot \gamma\rangle \vdash \cdots \vdash \langle f_{R_2}, \rho_n, f_R \cdot \gamma\rangle$ is a regular derivation of $\mathcal{P}_{R_2}$, then the following is a regular derivation of $\mathcal{P}_R$.*

$$\begin{array}{llll}
\langle i_R,\ \rho_0, & \gamma\rangle & \vdash \cdots \vdash \langle f_R,\ \rho_k, & \gamma\rangle \quad \vdash_{t5} \\
\langle i_{R_2}, \rho_k, f_R \cdot \gamma\rangle & \vdash \cdots \vdash & \langle f_{R_2}, \rho_n, f_R \cdot \gamma\rangle & \vdash_p \\
\langle f_R,\ \rho_n, & \gamma\rangle &&
\end{array}$$

We are left to show that each derivation of $\mathcal{P}_R$ that moves the PDA from the start state i_R to the final state f_R is regular and that regular derivations satisfy the required properties. In the following lemma, we show that derivations which leave a particular word γ at the bottom of the stack are regular and satisfy properties (1) and (2). Later, we will show that each accepting derivation of $\mathcal{P}_R$ is of this form.

Lemma 8.7. *For each role $R \in \mathsf{rol}_{\mathcal{R}}$, each word $\gamma \in \Gamma_{\mathcal{R}}^*$, and each derivation of the form $\langle \lambda_0, \rho_0, \gamma_0 \cdot \gamma\rangle \vdash \cdots \vdash \langle \lambda_n, \rho_n, \gamma_n \cdot \gamma\rangle$ with $\lambda_0 = i_R$, $\lambda_n = f_R$, and $\gamma_0 = \epsilon$,*

(i) the derivation is regular for $\mathcal{P}_R$;

(ii) for each $i \in [0..n]$, we have that $|\gamma_i| \leq 2 \cdot \mathsf{lv}(R)$; and

(iii) $R \Rightarrow^* \rho_0 - \rho_n$.

Proof. We prove the claim by induction on $n \in \mathbb{N}^+$.

Base case. For $n = 1$, consider an arbitrary role $R \in \mathsf{rol}_{\mathcal{R}}$, word $\gamma \in \Gamma_{\mathcal{R}}^*$, and sequence $\langle i_R, \rho_0, \gamma_0 \cdot \gamma\rangle \vdash \langle f_R, \rho_1, \gamma_1 \cdot \gamma\rangle$. By Definition 8.1, only transitions from cases (r), (t1),

(ur), and (u1) move $\mathcal{P}_R$ from state i_R to state f_R. These transitions leave the stack untouched, so $\gamma_1 = \epsilon = \gamma_0$ and property (ii) holds. For properties (i) and (iii), we next consider the four different forms that the sequence may take.

- $\langle i_R, R \cdot \rho_1, \gamma_0 \cdot \gamma\rangle \vdash_r \langle f_R, \rho_1, \gamma_1 \cdot \gamma\rangle$. Then $R \neq \top_r$, so this is a regular derivation of $\mathcal{P}_R$ by case s_r and (i) holds. Finally, $\rho_0 - \rho_1 = R$, which implies $R \Rightarrow^* \rho_0 - \rho_1$, and so (iii) holds.

- $\langle i_R, \rho_0, \gamma_0 \cdot \gamma\rangle \vdash_{t1} \langle f_R, \rho_0, \gamma_1 \cdot \gamma\rangle$. Then $R \neq \top_r$, so this is a regular derivation of $\mathcal{P}_R$ by case s_{t1} and (i) holds. Finally, $\rho_0 - \rho_1 = \epsilon$; moreover, by case t1 of Definition 8.1, we have $\epsilon \sqsubseteq R \in \mathcal{R}$, so $R \Rightarrow^* \epsilon$; hence, $R \Rightarrow^* \rho_0 - \rho_1$ and (iii) holds.

- $\langle i_R, T \cdot \rho_1, \gamma_0 \cdot \gamma\rangle \vdash_{ur} \langle f_R, \rho_1, \gamma_1 \cdot \gamma\rangle$. Then $R = \top_r$ and $T \in \mathsf{rol}_{\mathcal{R}}$, so this is a regular derivation of $\mathcal{P}_{\top_r}$ by case s_{ur} and (i) holds. Finally, $\rho_0 - \rho_1 = T \in \mathsf{rol}_{\mathcal{R}}$, which implies $R \Rightarrow^* \rho_0 - \rho_1$, and so (iii) holds.

- $\langle i_R, \rho_0, \gamma_0 \cdot \gamma\rangle \vdash_{u1} \langle f_R, \rho_0, \gamma_1 \cdot \gamma\rangle$. Then $R = \top_r$, so this is a regular derivation of $\mathcal{P}_{\top_r}$ by case s_{u1} and property (i) holds. Finally, $\rho_0 - \rho_1 = \epsilon$; hence, $R \Rightarrow^* \epsilon$, and so (iii) holds.

Inductive step. Consider an arbitrary $n \in \mathbb{N}^+$ and assume that (i)–(iii) hold for each role $R' \in \mathsf{rol}_{\mathcal{R}}$, each word $\gamma' \in \Gamma_{\mathcal{R}}^*$, and each derivation $\langle \lambda'_0, \rho'_0, \gamma'_0 \cdot \gamma'\rangle \vdash \cdots \vdash \langle \lambda'_c, \rho'_c, \gamma'_c \cdot \gamma'\rangle$ of length at most n and of the form required by this lemma. Furthermore, consider an arbitrary role $R \in \mathsf{rol}_{\mathcal{R}}$, an arbitrary word $\gamma \in \Gamma_{\mathcal{R}}^*$, and an arbitrary derivation

$$\langle \lambda_0,\, \rho_0,\, \gamma_0 \cdot \gamma\rangle \vdash \cdots \vdash \langle \lambda_{n+1},\, \rho_{n+1},\, \gamma_{n+1} \cdot \gamma\rangle \tag{8.7}$$

of length $n+1$ such that $\lambda_0 = i_R$, $\gamma_0 = \epsilon$, and $\lambda_{n+1} = f_R$. For each $i \in [0..n-1]$, let $x_i \in \{r, t1, \ldots, t5, ur, u1, u2, p\}$ be the form of derivation step i—that is, we fix x_i (arbitrarily if there is more than one possibility) such that $\langle \lambda_i, \rho_i, \gamma_i \cdot \gamma\rangle \vdash_{x_i} \langle \lambda_{i+1}, \rho_{i+1}, \gamma_{i+1} \cdot \gamma\rangle$ holds. We next consider the possible forms the sequence might have, and we show that properties (i)–(iii) hold in each case.

Case 1. $R = \top_r$. We start by considering the form of the first derivation step; that is, the form of $\langle \lambda_0, \rho_0, \gamma_0 \cdot \gamma \rangle \vdash_{x_0} \langle \lambda_1, \rho_1, \gamma_1 \cdot \gamma \rangle$. Given that $\lambda_0 = i_{\top_r}$, we have that $x_0 \in \{t1, t3, t4, ur, u1\}$. As $\mathcal{R}$ is normalised, each role inclusion $\rho \sqsubseteq R' \in \mathcal{R}$ is such that $R' \neq \top_r$, so $x_0 \in \{ur, u1\}$ and we have $\lambda_1 = f_{\top_r}$ and $\gamma_1 = \epsilon = \gamma_0$. Since $n > 1$, derivation step $\langle \lambda_1, \rho_1, \gamma_1 \cdot \gamma \rangle \vdash_{x1} \langle \lambda_2, \rho_1, \gamma_2 \cdot \gamma \rangle$ occurs in the sequence with $x_1 \in \{t2, t5, u2, p\}$. Since $\lambda_1 = f_{\top_r}$ and $\mathcal{R}$ is normalised, we have $x_1 \in \{u2, p\}$; furthermore, since $\gamma_1 = \epsilon$ and by our assumption on the form of (8.7), we have $x_1 \neq p$. Hence, the only remaining possibility is that $x_1 = u2$. By case (u2) in Definition 8.1, we have $\lambda_2 = i_{\top_r}$, $\rho_2 = \rho_1$, and $\gamma_2 = \gamma_1$. We next prove that properties (i)–(iii) hold.

(i) By s_{ur} and s_{u1} in Definition 8.6, we have that $\langle \lambda_0, \rho_0, \gamma_0 \cdot \gamma \rangle \vdash_{x_0} \langle \lambda_1, \rho_1, \gamma_1 \cdot \gamma \rangle$ is a regular derivation of $\mathcal{P}_{\top_r}$. By the inductive hypothesis, we also have that derivation $\langle \lambda_2, \rho_2, \gamma_2 \cdot \gamma \rangle \vdash \cdots \vdash \langle \lambda_{n+1}, \rho_{n+1}, \gamma_{n+1} \cdot \gamma \rangle$ is regular derivation for $\mathcal{P}_R$. By the definition of regular derivations, we have $\gamma_n = \gamma_2 = \epsilon$. But then, (8.7) is a regular derivation of $\mathcal{P}_R$ by case s_{u2}.

(ii) Words γ_0, γ_1, γ_2 are all empty. By the inductive hypothesis, we have $|\gamma_\ell| \leq 2 \cdot \mathsf{lv}(\top_r)$ for each $\ell \in [2..n+1]$. Thus, $|\gamma_i| \leq 2 \cdot \mathsf{lv}(\top_r)$ holds for each $i \in [0..n+1]$.

(iii) By the inductive hypothesis, we have $\top_r \Rightarrow^* \rho_2 - \rho_{n+1}$. By cases (ur) and (u1), either $\rho_0 - \rho_2 = \epsilon$ or $\rho_0 - \rho_2 = T \in \mathsf{rol}_{\mathcal{R}}$. But then, $\top_r \Rightarrow^* \rho_0 - \rho_{n+1}$ holds.

Case 2. $R \neq \top_r$ and an index $k \in [0..n]$ exists such that $\lambda_k = f_R$ and the k-th derivation step is of the form $\langle \lambda_k, \rho_k, \gamma_k \cdot \gamma \rangle \vdash_{t2} \langle \lambda_{k+1}, \rho_{k+1}, \gamma_{k+1} \cdot \gamma \rangle$. Then, by case (t2) in Definition 8.1, we have $R \cdot R \sqsubseteq R \in \mathcal{R}$, $\lambda_{k+1} = i_R$, $\rho_{k+1} = \rho_k$, and $\gamma_{k+1} = \gamma_k$. We next prove that properties (i)–(iii) hold.

(i) By the inductive hypothesis, $\langle \lambda_0, \rho_0, \gamma_0 \cdot \gamma \rangle \vdash \cdots \vdash \langle \lambda_k, \rho_k, \gamma_k \cdot \gamma \rangle$ is a regular derivation of $\mathcal{P}_R$. By the definition of regular derivations, we have $\gamma_k = \gamma_0 = \epsilon$. As $\lambda_{k+1} = i_R$ and $\gamma_{k+1} = \gamma_k = \epsilon$, we have $\langle \lambda_{k+1}, \rho_{k+1}, \gamma_{k+1} \cdot \gamma \rangle \vdash \cdots \vdash \langle \lambda_{n+1}, \rho_{n+1}, \gamma_{n+1} \cdot \gamma \rangle$ is of the form shown in (8.7) and it is shorter than $n+1$ so, by the inductive hypothesis, it is a regular derivation of $\mathcal{P}_R$. So (8.7) is a regular derivation of $\mathcal{P}_R$ by case s_{t2}.

(ii) By the inductive hypothesis, we have $|\gamma_{\ell_1}| \leq 2 \cdot \mathsf{lv}(R)$ for each $\ell_1 \in [0..k]$, as well as $|\gamma_{\ell_2}| \leq 2 \cdot \mathsf{lv}(R)$ for each $\ell_2 \in [k+1..n+1]$. Therefore, $|\gamma_i| \leq 2 \cdot \mathsf{lv}(R)$ holds for each $i \in [0..n+1]$.

(iii) By the inductive hypothesis, we have that $R \Rightarrow^* \rho_0 - \rho_k$ and $R \Rightarrow^* \rho_{k+1} - \rho_{n+1}$. But then, $R \cdot R \sqsubseteq R \in \mathcal{R}$ and $\rho_{k+1} = \rho_k$ implies that $R \Rightarrow^* \rho_0 - \rho_{n+1}$ holds.

Case 3. $R \neq \top_r$ and there is no $\ell \in [0..n]$ such that $\lambda_\ell = f_R$ and the ℓ-th derivation step is of the form $\langle \lambda_\ell, \rho_\ell, \gamma_\ell \cdot \gamma \rangle \vdash_{t2} \langle \lambda_{\ell+1}, \rho_{\ell+1}, \gamma_{\ell+1} \cdot \gamma \rangle$, but $k \in [0..n]$ exists such that $\langle \lambda_k, \rho_k, \gamma_k \cdot \gamma \rangle \vdash_{t5} \langle \lambda_{k+1}, \rho_{k+1}, \gamma_{k+1} \cdot \gamma \rangle$ and $\lambda_k = f_R$. Then, let k be the largest such index—that is, we assume that no $m > k$ exists such that $\langle \lambda_m, \rho_m, \gamma_m \cdot \gamma \rangle \vdash_{t5} \langle \lambda_{m+1}, \rho_{m+1}, \gamma_{m+1} \cdot \gamma \rangle$ and $\lambda_m = f_R$. Then, by case (t5) in Definition 8.1, for some role R_2 of level less than R, we have that $R \cdot R_2 \sqsubseteq R \in \mathcal{R}$, $\lambda_{k+1} = i_{R_2}$, $\rho_{k+1} = \rho_k$, and $\gamma_{k+1} = f_R \cdot \gamma_k$. We next prove that properties (i)–(iii) hold.

(i) As $\lambda_k = f_R$, by the inductive hypothesis, we have $\langle \lambda_0, \rho_0, \gamma_0 \cdot \gamma \rangle \vdash \cdots \vdash \langle \lambda_k, \rho_k, \gamma_k \cdot \gamma \rangle$ is a regular derivation of $\mathcal{P}_R$. By Definition 8.7, we have $\gamma_k = \gamma_0$. Since $\lambda_{k+1} = i_{R_2}$ and $\gamma_{k+1} = f_R \cdot \gamma_0$, by Lemma 8.5, an index $j \in [k+1..n]$ exists such that $\lambda_j = f_{R_2}$ and $\gamma_j = \gamma_{k+1}$; moreover, $\lambda_{j+1} = f_R$ and $\gamma_{j+1} = \gamma_0$ and $\rho_{j+1} = \rho_j$. We prove that $j + 1 = n + 1$. For the sake of contradiction, assume that $j + 1 < n + 1$ and consider the form of transition $\langle \lambda_{j+1}, \rho_{j+1}, \gamma_{j+1} \cdot \gamma \rangle \vdash_{x_{j+1}} \langle \lambda_{j+2}, \rho_{j+2}, \gamma_{j+2} \cdot \gamma \rangle$. Given that $\lambda_{j+1} = f_R$ and $R \neq \top_r$, we must have $x_{j+1} \in \{t2, t5, p\}$. By the initial assumption, we have $x_{j+1} \neq t2$; furthermore, by the maximality of k, we have $x_{j+1} \neq t5$; finally, since $\gamma_{j+1} = \gamma_0 = \epsilon$, we have $x_{j+1} \neq p$. Thus, $j + 1 = n + 1$, as required. It follows that the sequence is of the following form, where $\rho_{k+1} = \rho_k$ and $\rho_{n+1} = \rho_n$.

$$
\begin{array}{llll}
\langle i_R, & \rho_0, & \gamma_0 \cdot \gamma \rangle & \vdash \cdots \vdash \quad \langle f_R, \ \rho_k, \ \gamma_0 \cdot \gamma \rangle \quad \vdash_{t5} \\
\langle i_{R_2}, & \rho_{k+1}, & \gamma_{k+1} \cdot \gamma \rangle & \vdash \cdots \vdash \quad \langle f_{R_2}, \ \rho_n, \ \gamma_n \cdot \gamma \rangle \quad \vdash_{p} \\
\langle f_R, & \rho_{n+1}, & \gamma_0 \cdot \gamma \rangle &
\end{array}
$$

By Lemma 8.5, for each $\ell \in [k+1..n]$, we have that γ_ℓ is of the form $\gamma_\ell = \gamma''_\ell \cdot f_R \cdot \gamma_0$. In particular, words γ''_{k+1} and γ''_n are both empty. Then, by the inductive hypothesis,

we have that $\langle i_{R_2}, \rho_{k+1}, \gamma_{k+1} \cdot \gamma\rangle \vdash \cdots \vdash \langle f_{R_2}, \rho_n, \gamma_n \cdot \gamma\rangle$ is a regular derivation of $\mathcal{P}_{R_2}$. By case s_{t5}, then (8.7) is a regular derivation of $\mathcal{P}_R$.

(ii) By the inductive hypothesis, for each $\ell_1 \in [0..k]$, we have that $|\gamma_{\ell_1}| \leq 2 \cdot \mathsf{lv}(R)$. Furthermore, for each $\ell_2 \in [k+1..n]$, we have that $|\gamma''_{\ell_2}| \leq 2 \cdot \mathsf{lv}(R_2)$. Since $\mathsf{lv}(R_2) < \mathsf{lv}(R)$ and $\gamma_{\ell_2} = \gamma''_{\ell_2} \cdot f_R$, we also have that $|\gamma_{\ell_2}| \leq 2 \cdot \mathsf{lv}(R)$. Given that $\gamma_{n+1} = \epsilon$, for each $i \in [0..n+1]$, we have that $|\gamma_i| \leq 2 \cdot \mathsf{lv}(R)$.

(iii) By the inductive hypothesis, we have that $R \Rightarrow^* \rho_0 - \rho_k$ and $R_2 \Rightarrow^* \rho_{k+1} - \rho_n$. Given that $R \Rightarrow^* R \cdot R_2$, that $\rho_{k+1} = \rho_k$, and that $\rho_{n+1} = \rho_n$, we obtain that $R \Rightarrow^* \rho_0 - \rho_{n+1}$.

Case 4. $R \neq \top_r$ and no index $\ell \in [0..n]$ exists such that $\lambda_\ell = f_R$, $x_\ell \in \{t2, t5\}$, and $\langle \lambda_\ell, \rho_\ell, \gamma_\ell \cdot \gamma\rangle \vdash_{x_\ell} \langle \lambda_{\ell+1}, \rho_{\ell+1}, \gamma_{\ell+1} \cdot \gamma\rangle$; but we have $\langle \lambda_0, \rho_0, \gamma_0 \cdot \gamma\rangle \vdash_{t3} \langle \lambda_1, \rho_1, \gamma_1 \cdot \gamma\rangle$. Then, by case (t3) in Definition 8.1, for some role R_1 of level less than R, we have $R_1 \cdot R \sqsubseteq R \in \mathcal{R}$, $\lambda_1 = i_{R_1}$, $\rho_1 = \rho_0$, and $\gamma_1 = i_R \cdot \gamma_0$. We next prove that properties (i)–(iii) hold.

(i) Since $\lambda_1 = i_{R_1}$ and $\gamma_1 = i_R \cdot \gamma_0$, by Lemma 8.5, some $j \in [1..n]$ exists such that $\lambda_j = f_{R_1}$ and $\gamma_j = \gamma_1$; furthermore, $\lambda_{j+1} = i_R$ and $\gamma_{j+1} = \gamma_0$ and $\rho_{j+1} = \rho_j$. Then, the sequence is of the following form, where $\rho_1 = \rho_0$.

$$\begin{array}{llllll}
\langle i_R, & \rho_0, & \gamma_0 \cdot \gamma\rangle & \vdash_{t3} & & \\
\langle i_{R_1}, & \rho_1, & \gamma_1 \cdot \gamma\rangle & \vdash \cdots \vdash & \langle f_{R_1}, \rho_j, & \gamma_j \cdot \gamma\rangle \ \vdash_p \\
\langle i_R, & \rho_{j+1}, & \gamma_{j+1} \cdot \gamma\rangle & \vdash \cdots \vdash & \langle f_R, \ \rho_{n+1}, & \gamma_{n+1} \cdot \gamma\rangle
\end{array}$$

By Lemma 8.5, for each $\ell \in [1..j]$, word γ_ℓ is of the form $\gamma_\ell = \gamma''_\ell \cdot i_R \cdot \gamma_0$; in particular, we have that words γ''_1 and γ''_j are both empty. By the inductive hypothesis, we have that $\langle i_{R_1}, \rho_0, \gamma_1 \cdot \gamma\rangle \vdash \cdots \vdash \langle f_{R_1}, \rho_j, \gamma_j \cdot \gamma\rangle$ is a regular derivation of $\mathcal{P}_{R_1}$. As $\gamma_{j+1} = \gamma_0$, by the inductive hypothesis, then $\langle i_R, \rho_{j+1}, \gamma_{j+1} \cdot \gamma\rangle \vdash \cdots \vdash \langle f_R, \rho_{n+1}, \gamma_{n+1} \cdot \gamma\rangle$ is a regular derivation of $\mathcal{P}_R$. By case s_{t3}, then (8.7) is a regular derivation of $\mathcal{P}_R$.

(ii) By the inductive hypothesis, for each $\ell_2 \in [j+1..n+1]$, we have that $|\gamma_{\ell_2}| \leq 2 \cdot \mathsf{lv}(R)$; furthermore, for each $\ell_1 \in [1..j]$, we have that $|\gamma''_{\ell_1}| \leq 2 \cdot \mathsf{lv}(R_1)$. Since $\mathsf{lv}(R_1) < \mathsf{lv}(R)$ and $\gamma_{\ell_1} = \gamma''_{\ell_1} \cdot i_R$, we also have that $|\gamma_{\ell_1}| \leq 2 \cdot \mathsf{lv}(R)$. Finally, since $\gamma_0 = \epsilon$, for each $i \in [0..n+1]$, we have that $|\gamma_i| \leq 2 \cdot \mathsf{lv}(R)$.

(iii) By the inductive hypothesis, we have that $R_1 \Rightarrow^* \rho_1 - \rho_j$ and $R \Rightarrow^* \rho_{j+1} - \rho_{n+1}$. Given that $R \Rightarrow^* R_1 \cdot R$, that $\rho_1 = \rho_0$, and that $\rho_{j+1} = \rho_j$, we have $R \Rightarrow^* \rho_0 - \rho_{n+1}$.

Case 5. $R \neq \top_r$ and there is no index $\ell \in [0..n]$ such that $\lambda_\ell = f_R$, $x_\ell \in \{t2, t5\}$, and $\langle \lambda_\ell, \rho_\ell, \gamma_\ell \cdot \gamma \rangle \vdash_{x_\ell} \langle \lambda_{\ell+1}, \rho_{\ell+1}, \gamma_{\ell+1} \cdot \gamma \rangle$; in addition, $\langle \lambda_0, \rho_0, \gamma_0 \cdot \gamma \rangle \vdash_{x_0} \langle \lambda_1, \rho_1, \gamma_1 \cdot \gamma \rangle$ is such that $x_0 \neq t3$. We next consider the remaining possibilities for x_0. As $\lambda_0 = i_R$, we have that $x_0 \notin \{t2, t5, u2, p\}$ by cases (t2), (t5), (u2), and (p) of Definition 8.1; furthermore, due to $R \neq \top_r$, we have $x_0 \notin \{ur, u1\}$ by cases (ur) and (u1) of Definition 8.1. Moreover, assume that $x_0 \in \{r, t1\}$; then, we have $\lambda_1 = f_R$ and $\gamma_1 = \gamma_0$ by cases (r) and (t1) of Definition 8.1; since $n > 1$ and $R \neq \top_r$, the only possibility is that $\langle \lambda_1, \rho_1, \gamma_1 \cdot \gamma \rangle \vdash_p \langle \lambda_2, \rho_2, \gamma_2 \cdot \gamma \rangle$, which is impossible due to $\gamma_1 = \epsilon$ and our assumption on the form of (8.7). The only remaining possibility is that $x_0 = t4$. By case (t4) in Definition 8.1, one of the following holds.

- $\gamma_1 = f_R \cdot \gamma_0$ and a role R_1 with $\mathsf{lv}(R_1) < \mathsf{lv}(R)$ exist such that $R_1 \sqsubseteq R \in \mathcal{R}$, $\lambda_1 = i_{R_1}$, and $\rho_1 = \rho_0$.
- $\gamma_1 \neq f_R \cdot \gamma_0$, and R_1 and R_2 of level less than R exist such that $R_1 \cdot R_2 \sqsubseteq R \in \mathcal{R}$, $\lambda_1 = i_{R_1}$, $\rho_1 = \rho_0$, and $\gamma_1 = i_{R_2} \cdot f_R \cdot \gamma_0$.

We next consider the latter case, and show that properties (i)–(iii) hold; for the remaining case the properties can be proved similarly.

(i) Since $\lambda_1 = i_{R_1}$ and $\gamma_1 = i_{R_2} \cdot f_R \cdot \gamma_0$, by Lemma 8.5, $j_1 \in [1..n]$ exists such that $\lambda_{j_1} = f_{R_1}$ and $\gamma_{j_1} = \gamma_1$; moreover, $\lambda_{j_1+1} = i_{R_2}$ and $\gamma_{j_1+1} = f_R \cdot \gamma_0$ and $\rho_{j_1+1} = \rho_{j_1}$. By Lemma 8.5, we also have that $j_2 \in [j_1 + 1..n]$ exists such that $\lambda_{j_2} = f_{R_2}$ and $\gamma_{j_2} = \gamma_{j_1+1}$; furthermore, $\lambda_{j_2+1} = f_R$ and $\gamma_{j_2+1} = \gamma_0$ and $\rho_{j_2+1} = \rho_{j_2}$. Next, we prove that $j_2 + 1 = n + 1$. For the sake of contradiction, suppose that $j_2 + 1 < n + 1$ and consider the form of $\langle \lambda_{j_2+1}, \rho_{j_2+1}, \gamma_{j_2+1} \cdot \gamma \rangle \vdash_{x_{j_2+1}} \langle \lambda_{j_2+2}, \rho_{j_2+2}, \gamma_{j_2+2} \cdot \gamma \rangle$. Given that $\lambda_{j_2+1} = f_R$, we must have that $x_{j_2+1} \in \{t2, t5, u2, p\}$. However, we assumed that $x_{j_2+1} \notin \{t2, t5\}$ and that $R \neq \top_r$, so $x_{j_2+1} \neq u2$; finally, since $\gamma_{j_2+1} = \gamma_0 = \epsilon$, we have $x_{j_2+1} \neq p$. Therefore, we have $j_2 + 1 = n + 1$, as required, and the sequence is of the

following form, for $\rho_1 = \rho_0$, $\rho_{j_1+1} = \rho_j$, and $\rho_{n+1} = \rho_n$.

$$\begin{array}{llll} \langle i_R, & \rho_0, & \gamma_0 \cdot \gamma \rangle & \vdash_{t4} \\ \langle i_{R_1}, & \rho_1, & \gamma_1 \cdot \gamma \rangle & \vdash \cdots \vdash \langle f_{R_1}, \rho_{j_1}, \gamma_{j_1} \cdot \gamma \rangle \vdash_p \\ \langle i_{R_2}, & \rho_{j_1+1}, & \gamma_{j_1+1} \cdot \gamma \rangle & \vdash \cdots \vdash \langle f_{R_2}, \rho_n, \gamma_n \cdot \gamma \rangle \vdash_p \\ \langle f_R, & \rho_{n+1}, & \gamma_{n+1} \cdot \gamma \rangle & \end{array}$$

By Lemma 8.5, for each $\ell_1 \in [1..j_1]$, word γ_{ℓ_1} is of the form $\gamma_{\ell_1} = \gamma''_{\ell_1} \cdot i_{R_2} \cdot f_R \cdot \gamma_0$; in particular, we have that words γ''_1 and γ''_{j_1} are both empty. Then, by the inductive hypothesis, we have that $\langle i_{R_1}, \rho_1, \gamma_1 \cdot \gamma \rangle \vdash \cdots \vdash \langle f_{R_1}, \rho_{j_1}, \gamma_{j_1} \cdot \gamma \rangle$ is a regular derivation of $\mathcal{P}_{R_1}$. Similarly, by Lemma 8.5, for each $\ell_2 \in [j_1 + 1..n]$, we have that γ_{ℓ_2} is of the form $\gamma_{\ell_2} = \gamma''_{\ell_2} \cdot f_R \cdot \gamma_0$. In particular, words γ''_{j_1+1} and γ''_n are both empty. By the inductive hypothesis, then $\langle i_{R_2}, \rho_{j_1+1}, \gamma_{j_1+1} \cdot \gamma \rangle \vdash \cdots \vdash \langle f_{R_2}, \rho_n, \gamma_n \cdot \gamma \rangle$ is a regular derivation of $\mathcal{P}_{R_2}$. By case s_{t4}, then (8.7) is a regular derivation of $\mathcal{P}_R$.

(ii) By the inductive hypothesis, for each $\ell_1 \in [1..j_1]$, we have that $|\gamma''_{\ell_1}| \leq 2 \cdot \mathsf{lv}(R_1)$. Since $\mathsf{lv}(R_1) < \mathsf{lv}(R)$ and $\gamma_{\ell_1} = \gamma''_{\ell_1} \cdot i_{R_2} \cdot f_R$, we also have that $|\gamma_{\ell_1}| \leq 2 \cdot \mathsf{lv}(R)$. Similarly, by the inductive hypothesis, for each $\ell_2 \in [j_1 + 1..n]$, we have that $|\gamma''_{\ell_2}| \leq 2 \cdot \mathsf{lv}(R_2)$. Since $\mathsf{lv}(R_2) < \mathsf{lv}(R)$ and $\gamma_{\ell_2} = \gamma''_{\ell_2} \cdot f_R$, we also have $|\gamma_{\ell_2}| \leq 2 \cdot \mathsf{lv}(R)$. Since $\gamma_0 = \epsilon$, for each $i \in [0..n + 1]$, we have that $|\gamma_i| \leq 2 \cdot \mathsf{lv}(R)$.

(iii) By the inductive hypothesis, we have that $R_1 \Rightarrow^* \rho_1 - \rho_{j_1}$ and $R_2 \Rightarrow^* \rho_{j_1+1} - \rho_n$. Given that $R \Rightarrow^* R_1 \cdot R_2$, that $\rho_1 = \rho_0$, and that $\rho_{n+1} = \rho_n$, we conclude that $R \Rightarrow^* \rho_0 - \rho_{n+1}$.

There are no other possibilities for the form of (8.7), so the claim of this lemma holds for each derivation of that form. □

We are finally ready to show that PDA $\mathcal{P}_R$ satisfies properties (1) and (2).

Lemma 8.8. *For each role $R \in \mathsf{rol}_\mathcal{R}$ and each role chain $\rho \in (\mathsf{rol}_\mathcal{R})^*$,*

1. *$\rho \in \mathcal{L}(\mathcal{P}_R)$ implies $\rho \in \mathcal{L}(R)$, and*
2. *$\mathcal{P}_R$ has stack bounded by $2 \cdot \mathsf{lv}(R) + 1$.*

Table 8.1: Definition of derivation (8.9) depending on the form of axiom $\rho \sqsubseteq T$

Type	Axiom	Derivation
(t1)	$\epsilon \sqsubseteq T$	$\langle i_T, \rho'', \gamma_i\rangle \vdash_{t1} \langle f_T, \rho'', \gamma_i\rangle$
(t2)	$T \cdot T \sqsubseteq T$	$\langle i_T, T \cdot T \cdot \rho'', \gamma_i\rangle \vdash_r \langle f_T, T \cdot \rho'', \gamma_i\rangle \vdash_{t2}$ $\langle i_T, T \cdot \rho'', \gamma_i\rangle \vdash_r \langle f_T, \rho'', \gamma_i\rangle$
(t3)	$T_1 \cdot T \sqsubseteq T$	$\langle i_T, T_1 \cdot T \cdot \rho'', \gamma_i\rangle \vdash_{t3}$ $\langle i_{T_1}, T_1 \cdot T \cdot \rho'', i_T \cdot \gamma_i\rangle \vdash_r \langle f_{T_1}, T \cdot \rho'', i_T \cdot \gamma_i\rangle \vdash_p$ $\langle i_T, T \cdot \rho'', \gamma_i\rangle \vdash_r \langle f_T, \rho'', \gamma_i\rangle$
(t4)	$T_1 \sqsubseteq T$	$\langle i_T, T_1 \cdot \rho'', \gamma_i\rangle \vdash_{t4}$ $\langle i_{T_1}, T_1 \cdot \rho'', f_T \cdot \gamma_i\rangle \vdash_r \langle f_{T_1}, \rho'', f_T \cdot \gamma_i\rangle \vdash_p$ $\langle f_T, \rho'', \gamma_i\rangle$
(t4)	$T_1 \cdot T_2 \sqsubseteq T$	$\langle i_T, T_1 \cdot T_2 \cdot \rho'', \gamma_i\rangle \vdash_{t4}$ $\langle i_{T_1}, T_1 \cdot T_2 \cdot \rho'', i_{T_2} \cdot f_T \cdot \gamma_i\rangle \vdash_r \langle f_{T_1}, T_2 \cdot \rho'', i_{T_2} \cdot f_T \cdot \gamma_i\rangle \vdash_p$ $\langle i_{T_2}, T_2 \cdot \rho'', f_T \cdot \gamma_i\rangle \vdash_r \langle f_{T_2}, \rho'', f_T \cdot \gamma_i\rangle \vdash_p$ $\langle f_T, \rho'', \gamma_i\rangle$
(t5)	$T \cdot T_2 \sqsubseteq T$	$\langle i_T, T \cdot T_2 \cdot \rho'', \gamma_i\rangle \vdash_r \langle f_T, T_2 \cdot \rho'', \gamma_i\rangle \vdash_{t5}$ $\langle i_{T_2}, T_2 \cdot \rho'', f_T \cdot \gamma_i\rangle \vdash_r \langle f_{T_2}, \rho'', f_T \cdot \gamma_i\rangle \vdash_p$ $\langle f_T, \rho'', \gamma_i\rangle$

Proof. By the definition of $\mathcal{P}_R$, transitions resulting from case (p) in Definition 8.1 are the only ones popping elements from the stack, and these never pop symbol $\perp$; hence, at each point i in an accepting derivation of $\mathcal{P}_R$, the stack content γ_i is of the form $\gamma_i = \gamma_i' \cdot \perp$. Then, the two claims follow immediately from Lemma 8.7. □

8.2.2 Completeness

We next prove that our encoding is also complete, thus proving Theorem 8.3.

Lemma 8.9. *For each role* $R \in \mathsf{rol}_{\mathcal{R}}$ *and each role chain* $\rho \in (\mathsf{rol}_{\mathcal{R}})^*$*, if* $\rho \in \mathcal{L}(R)$ *then* $\rho \in \mathcal{L}(\mathcal{P}_R)$*.*

Proof. Consider an arbitrary role $R \in \mathsf{rol}_{\mathcal{R}}$. In the following, for each role chain ρ, we write $R \overset{0}{\Rightarrow} \rho$ if $\rho = R$; furthermore, for each $m \in \mathbb{N}^+$, we write $R \overset{m}{\Rightarrow} \rho$ if role chains $\rho_1, \ldots, \rho_m$ with $\rho_m = \rho$ exist such that $R \Rightarrow \rho_1 \Rightarrow \cdots \Rightarrow \rho_m$. By the definition of $\mathcal{L}(R)$, we have that $\rho \in \mathcal{L}(R)$ if and only if a natural number $m \in \mathbb{N}$ exists such that $R \overset{m}{\Rightarrow} \rho$. By induction on $m \in \mathbb{N}$, we next show that $R \overset{m}{\Rightarrow} \rho$ implies $\rho \in \mathcal{L}(\mathcal{P}_R)$.

Base case. Let $m = 0$. Then, we have that $R \overset{0}{\Rightarrow} R$. We consider two cases depending on the form of role $R \in \mathsf{rol}_{\mathcal{R}}$, and show that in each case we have that $R \in \mathcal{L}(\mathcal{P}_R)$, as required.

- $R = \top_r$. By case (ur) in Definition 8.1, we have that $\langle i_{\top_r}, \top_r, \bot\rangle \vdash_{ur} \langle f_{\top_r}, \epsilon, \bot\rangle$, so $R \in \mathcal{L}(\mathcal{P}_R)$.
- $R \in \mathsf{rol}_{\mathcal{R}} \setminus \{\top_r\}$. By case (r) in Definition 8.1, we have that $\langle i_R, R, \bot\rangle \vdash_r \langle f_R, \epsilon, \bot\rangle$, so $R \in \mathcal{L}(\mathcal{P}_R)$.

Inductive step. Consider an arbitrary $m \in \mathbb{N}$ and assume that, for each role chain ρ' such that $R \overset{m}{\Rightarrow} \rho'$, we have $\rho' \in \mathcal{L}(\mathcal{P}_R)$; we show that the same holds for $m + 1$. Then, consider arbitrary role chains $\rho_1, \ldots, \rho_{m+1}$ such that $R \Rightarrow \rho_1 \Rightarrow \cdots \Rightarrow \rho_m \Rightarrow \rho_{m+1}$. By the definition of relation $\Rightarrow$, a role $T \in \mathsf{rol}_{\mathcal{R}}$ and role chains ρ', ρ, ρ'' exist such that role chain ρ_m is of the form $\rho_m = \rho' \cdot T \cdot \rho''$, role chain ρ_{m+1} is of the form $\rho_{m+1} = \rho' \cdot \rho \cdot \rho''$, and $T \Rightarrow \rho$. Since $R \overset{m}{\Rightarrow} \rho' \cdot T \cdot \rho''$, by the inductive hypothesis, we have $\rho' \cdot T \cdot \rho'' \in \mathcal{L}(\mathcal{P}_R)$, so a sequence $\langle \lambda_0, \rho_0, \gamma_0\rangle \vdash \cdots \vdash \langle \lambda_n, \rho_n, \gamma_n\rangle$ of $\mathcal{P}_R$ exists with $\lambda_0 = i_R$ and $\lambda_n = f_R$; furthermore, $\gamma_0 = \bot$ and $\gamma_n = \bot$; finally, $\rho_0 = \rho' \cdot T \cdot \rho''$ and $\rho_n = \epsilon$. Then there exists an index $i \in [0..n-1]$ such that $\rho_i = T \cdot \rho''$ and $\rho_{i+1} = \rho''$. Furthermore, for each $j \in [0..i]$, role chain ρ_j is of the form $\rho_j := \rho_j^- \cdot T \cdot \rho''$ for some role chain $\rho_j^- \in \mathsf{rol}^*_{\mathcal{R}}$. Next, consider the form of x_i in $\langle \lambda_i, \rho_i, \gamma_i\rangle \vdash_{x_i} \langle \lambda_{i+1}, \rho_{i+1}, \gamma_{i+1}\rangle$. By Definition 8.1, only transitions in cases (r) and (ur) read symbols from the input, so $x_i \in \{r, ur\}$. We show that the lemma holds in each case.

Case 1. Consider the case in which $x_i = r$. Then, we have $\lambda_i = i_T$ and $\lambda_{i+1} = f_T$, $\gamma_i = \gamma_{i+1}$, and $T \in \mathsf{rol}_{\mathcal{R}} \setminus \{\top_r\}$. Due to $T \Rightarrow^* \rho$ and $T \neq \top_r$, we have $\rho \sqsubseteq T \in \mathcal{R}$. Then, the following is a derivation of $\mathcal{P}_R$

$$\langle \lambda_0, \rho_0^- \cdot \rho \cdot \rho'', \gamma_0\rangle \vdash \cdots \vdash \langle \lambda_i, \rho_i^- \cdot \rho \cdot \rho'', \gamma_i\rangle \vdash^* \tag{8.8}$$

$$[\text{The derivation from Table 8.1 for } \rho \sqsubseteq T] \vdash^* \tag{8.9}$$

$$\langle \lambda_{i+1},\ \rho'',\ \gamma_{i+1}\rangle \vdash \cdots \vdash \langle \lambda_n,\ \epsilon,\ \gamma_n\rangle \tag{8.10}$$

where the derivation in (8.9) is defined in Table 8.1 depending on the form of $\rho \sqsubseteq T \in \mathcal{R}$.

Case 2. Consider the case in which $x_i = ur$. Then, we have that $\lambda_i = i_{\top_r}$ and

$\lambda_{i+1} = f_{\top_r}$, $\gamma_i = \gamma_{i+1}$ and $T \in \mathsf{rol}_{\mathcal{R}}$. Then, the following is also a derivation of $\mathcal{P}_R$

$$\langle \lambda_0, \rho_0^- \cdot \rho \cdot \rho'', \gamma_0 \rangle \vdash \cdots \vdash \langle \lambda_i, \rho_i^- \cdot \rho \cdot \rho'', \gamma_i \rangle \vdash^* \tag{8.11}$$

$$[\text{The derivation } \Pi(\rho)] \vdash^* \tag{8.12}$$

$$\langle \lambda_{i+1}, \rho'', \gamma_{i+1} \rangle \vdash \cdots \vdash \langle \lambda_n, \epsilon,\ \gamma_n \rangle \tag{8.13}$$

where the derivation $\Pi(\rho)$ in (8.12) is inductively defined as follows.

$$\Pi(\rho) = \begin{cases} \langle i_{\top_r}, \rho'', \gamma_i \rangle \vdash_{u1} \langle f_{\top_r}, \rho'', \gamma_i \rangle & \text{if } \rho = \epsilon, \\ \langle i_{\top_r}, \rho \cdot \rho'', \gamma_i \rangle \vdash_{ur} \langle f_{\top_r}, \bar{\rho} \cdot \rho'', \gamma_i \rangle \vdash_{u2} \Pi(\bar{\rho}) & \text{if } \rho = P \cdot \bar{\rho}. \end{cases}$$

Therefore, in either case, we have $\rho' \cdot \rho \cdot \rho'' \in \mathcal{L}(\mathcal{P}_R)$, as required. □

Chapter 9

Answering CQs over $\mathcal{ELRO}_{\bot}^{+}$ KBs

Even though the datalog program $\mathsf{D}_{\mathcal{K}}$ of an $\mathcal{ELRO}_{\bot}^{+}$ knowledge base $\mathcal{K}$ can be used to check the consistency of $\mathcal{K}$, evaluating a conjunctive query q over $\mathsf{D}_{\mathcal{K}}$ can generate unsound answers. In Chapter 6, we presented an NP filtering procedure that checks whether a candidate answer τ to q over $\mathsf{D}_{\mathcal{K}}$ is sound, provided that $\mathcal{K}$ is in $\mathcal{ELHO}_{\bot}^{+}$. In this chapter, we use the encoding of regular role inclusions based on bounded-stack PDAs introduced in Chapter 8 to lift this restriction and present a PSPACE filtering procedure $\mathsf{isSound}_{RI}$ for $\mathcal{ELRO}_{\bot}^{+}$ that extends and refines the procedure for $\mathcal{ELHO}_{\bot}^{+}$. In Section 9.1 we discuss the key differences between the two filtering procedures; and in Section 9.2 we introduce the procedure $\mathsf{isSound}_{RI}$ formally and show that it can be implemented to use space polynomial in combined complexity. Similarly to the procedure for $\mathcal{ELHO}_{\bot}^{+}$, the filtering procedure $\mathsf{isSound}_{RI}$ runs in polynomial time if τ does not map query variables to auxiliary constants, or if τ does not map binary atoms in q to atoms in $\mathsf{D}_{\mathcal{K}}$ that are obtained using the complex role inclusions in $\mathcal{K}$. Furthermore, we show that our filtering procedure can be used to obtain the first worst-case optimal algorithm for answering CQs over $\mathcal{ELRO}_{\bot}^{+}$ knowledge bases that runs in PSPACE; thus, we settle the question of the combined complexity of CQ answering for OWL 2 EL. In Chapter 10, we shall show that CQ answering over $\mathcal{ELRO}_{\bot}^{+}$ knowledge bases is PSPACE-hard even when the query is fixed; hence, our algorithm is optimal also in knowledge base complexity. Finally, in Section 6.4 we prove the correctness of the filtering procedure.

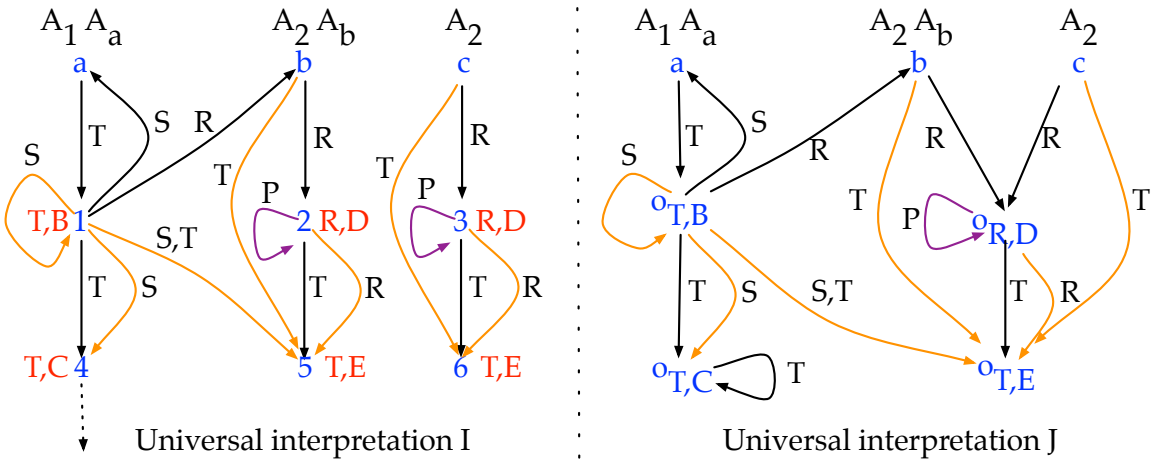

Figure 9.1: The universal interpretations I and J of $\Xi_{\mathcal{K}}$ and $\mathsf{D}_{\mathcal{K}}$, respectively

9.1 Intuition

Algorithm 2 on page 153 specifies a procedure $\mathsf{isSound}_{RI}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ that checks whether candidate answer τ to q over $\mathsf{D}_{\mathcal{K}}$ is sound. We explain the fundamental differences between the filtering procedure in Algorithm 2 and the filtering procedure for $\mathcal{ELHO}^{+}_{\bot}$ from Chapter 6 using the knowledge base $\mathcal{K}$ from Example 9.1.

Example 9.1. *Let $\mathcal{K}$ be the $\mathcal{ELRO}^{+}_{\bot}$ KB where the ABox $\mathcal{A} = \{A_1(a), A_2(b), A_2(c)\}$, and the TBox $\mathcal{T}$ and the RBox $\mathcal{R}$ contain the following axioms.*

$$
\begin{array}{lll}
A_1 \sqsubseteq \exists T.B & B \sqsubseteq \exists T.C & R \cdot T \sqsubseteq T \\
B \sqsubseteq \exists S.A_a & C \sqsubseteq \exists T.C & S \cdot T \sqsubseteq S \\
A_a \sqsubseteq \{a\} & A_2 \sqsubseteq \exists R.D & P \cdot T \sqsubseteq R \\
B \sqsubseteq \exists R.A_b & D \sqsubseteq \exists P.\mathsf{Self} & \\
A_b \sqsubseteq \{b\} & D \sqsubseteq \exists T.E &
\end{array}
$$

Moreover, let $\Xi_{\mathcal{K}}$ be the rule base for $\mathcal{K}$ and let $\mathsf{D}_{\mathcal{K}}$ be the datalog program for $\mathcal{K}$. Figure 9.1 shows the universal interpretations I and J of $\Xi_{\mathcal{K}}$ and $\mathsf{D}_{\mathcal{K}}$, respectively; notation is as in Example 4.1 on page 39. Axiom $C \sqsubseteq \exists T.C$ makes I infinite, so a terminating algorithm cannot simply materialise I and evaluate conjunctive queries in it. Our filtering procedure uses the PDA encoding of the RBox described in Chapter 8. The transition function $\delta_{\mathcal{R}}$ is

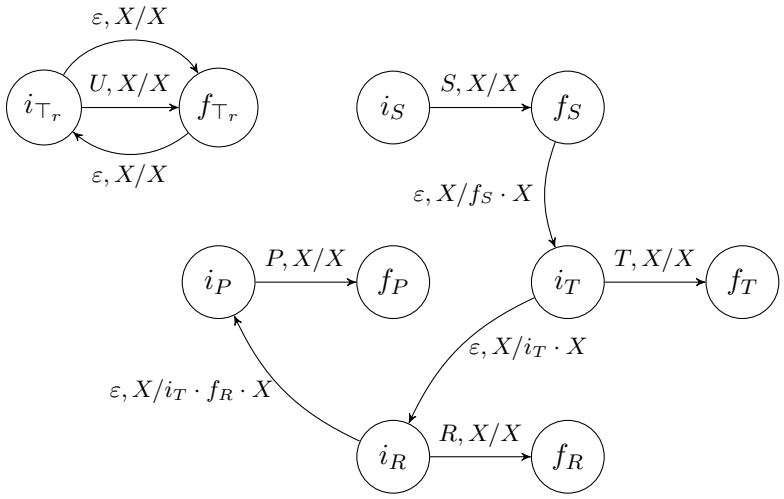

Figure 9.2: The transitions of $\delta_{\mathcal{R}}$ where $X \in \Gamma_{\mathcal{R}}$ and $U \in \mathsf{rol}_{\mathcal{R}}$

shown in Figure 9.2; notation is the same as in Example 8.2 on page 125.

The initial steps of our filtering procedure $\mathsf{isSound}_{RI}$ follow the initial steps of the filtering procedure for $\mathcal{ELHO}_{\bot}^{+}$. In step 1, we use the auxiliary function $\mathsf{isDSound}$ from Definition 6.5 to compute equality constraints using the aux-simple 'forks' in q, we check that these constraints are satisfied by q and τ and compute a new query $q_{\sim}$, and then we check that $q_{\sim}$ does not contain aux-simple cycles. When $q_{\sim}$ contains only atoms that are either good or aux-simple, step 1 suffices to determine whether the candidate answer τ is sound. Otherwise, in lines 3–6 we nondeterministically compute a variable renaming σ (see Definition 6.6) and a skeleton $\mathcal{S}$ for σ, q, and τ (see Definition 6.7). The skeleton is a finite structure that describes the (possibly infinite) set of all substitutions π mapping the variables of $\sigma(q_{\sim})$ to distinct labelled nulls of I. As we have seen in Chapter 6, the constraints expressed by the skeleton do not sufficiently describe the substitutions that map onto I the binary atoms of $\sigma(q_{\sim})$ that are neither good nor aux-simple, so additional steps are required to ensure that each substitution π represented by the skeleton correctly maps the query into the universal interpretation.

In the filtering procedure for $\mathcal{ELHO}_{\bot}^{+}$, we refine the skeleton's constraints by guessing, for each binary atom in the query that is neither good nor aux-simple, how to unfold it as a sequence of direct edges in I using the transitive roles in $\mathcal{K}$. We represent the unfolding of each binary atom by labelling each skeleton edge with a set of roles, and each skeleton

root with a set of pairs consisting of a skeleton vertex and a role. However, this unfolding step is based on properties that are particular to the universal interpretations of $\mathcal{ELHO}_\bot^+$ knowledge bases and cannot be immediately lifted to those of $\mathcal{ELRO}_\bot^+$ knowledge bases.

When $\mathcal{K}$ is an $\mathcal{ELHO}_\bot^+$ knowledge base, each edge $\langle w, w' \rangle$ in the universal interpretation I of $\mathcal{K}$ that is labelled by a role R can be unfolded either via a nominal path or via an anonymous path. Nominal paths contain at least one direct edge pointing towards an individual from $N_\mathfrak{I}$, whereas anonymous paths consist only of direct edges pointing towards labelled nulls. Moreover, the relative position of the terms w and w' in I completely determines the type of the unfolding: if w reaches w' in I using an anonymous path then the edge $\langle w, w' \rangle$ will be unfolded using an anonymous path; otherwise $\langle w, w' \rangle$ will be unfolded using a nominal path. Finally, each unfolding, regardless of its type, is *uniform*: one can find a subrole P of R that labels each direct edge along the path.

Regular role inclusions in the knowledge base complicate the structure of the universal interpretation, so the aforementioned properties do not hold when $\mathcal{K}$ is an $\mathcal{ELRO}_\bot^+$ knowledge base. In particular, the relative positions of the terms in the universal interpretation I do not anymore determine the the type of the unfolding. For example, the universal interpretation I shown in Figure 9.1 contains the edge $S(1, 4)$ and 1 is directly connected to 4 in I; in spite of this, the only possible unfolding for $S(1, 4)$ is the nominal path $\rho_S = S \cdot T \cdot T$ that connects 1 to individual a, a back to 1, and 1 to 4. Moreover, each edge $\langle w, w' \rangle$ of I must be unfolded either via a nominal path or a *self-anonymous path*—a repeating sequence of zero or more self edges, followed by a direct edge pointing towards a labelled null, ending with zero or more self edges. Moreover, an unfolding need not be uniform: each edge along the path can be labelled by a different role. For example, the edge $R(2, 5)$ can be unfolded into the self-anonymous path $\rho_R = P \cdot T$ connecting 2 with itself using the self edge labelled by role P, and then connecting 2 with 5 using the direct edge labelled by role T.

Therefore, in the filtering procedure $\mathsf{isSound}_{RI}$, we represent the unfolding ρ_R of each binary atom $R(s, t)$ that is neither good nor aux-simple by labelling edges and root vertices of $\mathcal{S}$ with bounded-stack PDAs with transition function $\delta_\mathcal{R}$; this allows us to compactly capture each possible, non-uniform unfolding ρ_R. In particular, we account for the two types of unfolding using the guessing in step 9: if v_0 is a root of the skeleton, then ρ_R is

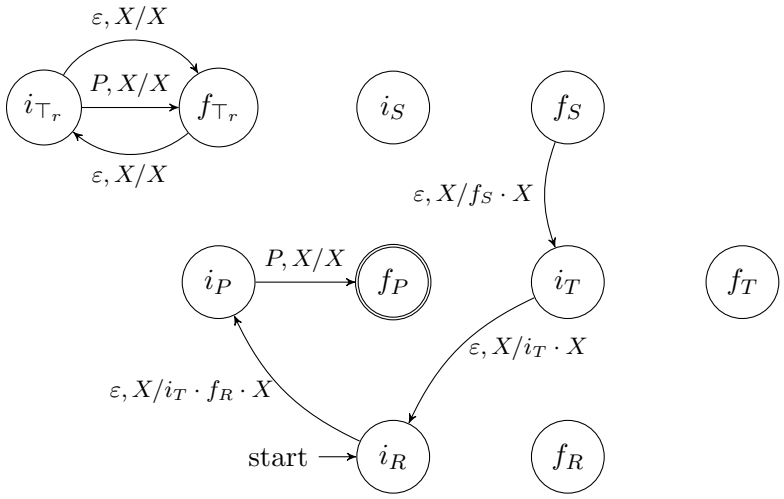

Figure 9.3: The stationary PDA $\mathsf{spda}(i_R, \bot, \lambda_0, \gamma_0, o_{R,D})$ where $X \in \Gamma_{\mathcal{R}}$

a nominal path; otherwise ρ_R is a self-anonymous path. Similar to the filtering procedure for $\mathcal{ELHO}^+_\bot$, in steps 10–12, we split the unfolding ρ_R into its subcomponents along the path $v_0, \ldots, v_n$ that connects v_0 to $v_n = t$ in the skeleton $\mathcal{S}$. To this end, we 'split' the PDA $\mathcal{P}_R$ encoding $\mathcal{L}(R)$ into n bounded-stack PDAs by nondeterministically guessing a sequence of states $\lambda_0, \ldots, \lambda_n$ from $Q_{\mathcal{R}}$ and stack words $\gamma_0, \ldots, \gamma_n$ over $\Gamma_{\mathcal{R}}$ such that the size of each γ_i is bounded by the depth of $\mathcal{R}$, and $\lambda_n = f_R$ and $\gamma_n = \bot$ are the final state and stack of $\mathcal{P}_R$, respectively. Then, for each $i \in [1..n]$, we construct the bounded-stack PDA $\mathsf{pda}(\lambda_{i-1}, \gamma_{i-1}, \lambda_i, \gamma_i)$ with transition function $\delta_{\mathcal{R}}$ such that λ_{i-1} and γ_{i-1} are the initial state and stack, and λ_i and γ_i are the final state and stack of the PDA, respectively. We use this PDA to describe the subpaths of ρ_R that connect $\pi(v_{i-1})$ to $\pi(v_i)$ in I; so we label the corresponding skeleton edge $\langle v_{i-1}, v_i \rangle \in \mathcal{S}$ with $\mathsf{pda}(\lambda_{i-1}, \gamma_{i-1}, \lambda_i, \gamma_i)$.

At this point, the skeleton describes the part of the unfolding ρ_R that connects $\pi(v_0)$ to $\pi(v_n) = \pi(t)$ in I and moves the PDA $\mathcal{P}_R$ from state λ_0 with stack γ_0 to the final state $\lambda_n = f_R$ with final stack $\gamma_n = \bot$. However, it does not describe the, possibly empty, initial part of ρ_R that connects $\pi(s)$ to $\pi(v_0)$ and moves $\mathcal{P}_R$ from its initial state i_R and stack $\bot$ to state λ_0 with stack γ_0. Hence, in steps 14–17, we consider two cases.

When v_0 is a skeleton root and so ρ_R is a nominal path, the initial part of ρ_R must connect $\pi(s)$ to the individual represented by v_0 in I. However, if v_0 is a placeholder variable, we do not know which individual of I v_0 represents, so we cannot check the

existence of the required subpath independently. Therefore, we add in step 15 the pair $\langle s, \mathsf{pda}(i_R, \bot, \lambda_0, \gamma_0)\rangle$ as a constraint to the root v_0, where $\mathsf{pda}(i_R, \bot, \lambda_0, \gamma_0)$ is the PDA with transition function $\delta_{\mathcal{R}}$ such that i_R and $\bot$ are the initial state and stack, and λ_0 and γ_0 are the final state and stack of the PDA, respectively; we use this PDA to describe the subpaths of ρ_R that connect $\pi(s)$ to $\pi(v_0)$ in I.

In contrast, when $v_0 = s$ is not a root and so ρ_R is a self-anonymous path, the initial part of ρ_R must consist of a self loop. In addition, in this case, we also know that v_0 is mapped by τ to an auxiliary constant, say $\tau(v_0) = o_{R,D}$. Hence, we capture the initial subpath of ρ_0 that moves $\mathcal{P}_R$ from its initial state i_R and stack $\bot$ to state λ_0 with stack γ_0 using a so-called *stationary PDA* $\mathsf{spda}(i_R, \bot, \lambda_0, \gamma_0, \tau(s))$. This stationary PDA describes all the possible loops on elements in I that have type R, D since $\tau(s) = o_{R,D}$; that is, $\rho \in \mathcal{L}(\mathsf{spda}(i_R, \bot, \lambda_0, \gamma_0, \tau(s)))$ for each element w in I of type R, D and for each role chain ρ corresponding to a self loop on w that moves the PDA from state i_R with stack $\bot$ to state λ_0 with stack γ_0. Because τ already determines the type of $\pi(s)$, we check directly in step 17 that the required loop can be realised by the universal interpretation I: using polynomial time we check that the language accepted by $\mathcal{L}(\mathsf{spda}(i_R, \bot, \lambda_0, \gamma_0, \tau(s)))$ is non-empty [50]. Figure 9.3 shows the stationary PDA $\mathsf{spda}(i_R, \bot, f_P, \gamma_0, o_{R,D})$ for an arbitrary stack word γ_0. In the universal interpretation I shown in Figure 9.1 the self edges that loop over elements in I of type R, D are all labelled by role P, so the transition function of $\mathsf{spda}(i_R, \bot, f_P, \gamma_0, o_{R,D})$ is obtained from $\delta_{\mathcal{R}}$ by removing each transition that consumes an input symbol different from P.

After steps 5–17, each substitution π that satisfies the skeleton's constraints maps $\sigma(q_\sim)$ onto I, so we are left to check that at least one such substitution can be realised by the universal interpretation I. Because the universal interpretation I of $\mathcal{K}$ can be infinite, in the filtering procedure for $\mathcal{ELHO}^+_\bot$ we reduced this task to checking the existence of paths in the universal interpretation J of $\mathsf{D}_\mathcal{K}$ that consist only of direct edges and are labelled with a set of roles. For $\mathcal{ELRO}^+_\bot$ knowledge bases the reduction is more involved, because we must check the existence of paths in J that are recognised by PDAs. In particular, we show that a substitution π exists that satisfies the skeleton's constraints if and only if the following two properties hold for each skeleton edge $\langle v, v'\rangle$ and each root vertex v_r of $\mathcal{S}$.

(a) For each PDA $\mathcal{P} \in L(v, v')$, a non-empty role chain $\rho \in \mathcal{L}(\mathcal{P})$ exists labelling a path from u to u' in J that starts with a direct edge and consists only of direct and self edges pointing to auxiliary constants, where u and u' are the constants representing $\pi(v)$ and $\pi(v')$ in J, respectively.

(b) For each $\langle s, \mathcal{P} \rangle \in L(v_r)$, a role chain $\rho \in \mathcal{P}$ exists labelling a path from u_s to u_r in J, where u_s and u_r are the constants representing $\pi(s)$ and $\pi(v_r)$ in J, respectively.

These properties provide us with an effective way of checking the skeleton's constraints using the direct and self predicates occurring in the datalog program $\mathsf{D}_\mathcal{K}$ by applying in steps 18–22 the procedure check_{RI} from Definition 9.4 to each skeleton vertex $v \in \mathcal{V}$.

We check property (a) in function check_{RI} by applying to each edge $\langle v, v' \rangle$ of $\mathcal{S}$ the auxiliary procedure $\mathsf{exist}_{RI}(u, u', L(v, v'))$ from Algorithm 3 on page 153. Roughly speaking, we run all PDAs in parallel in steps 7–14 of Algorithm 3. In step 8 we check existence of direct edges pointing towards auxiliary constants in J via entailment checking; after that, for each PDA, in step 10 we guess a state λ and a stack γ of the PDA, in step 11 we check whether the PDA can perform the move, and in step 14 we actually move the PDA. Due to self-restrictions and reflexive roles, however, the PDAs need not move in synchrony: after each move over a direct edge, each of the PDAs can independently loop on the current constant. To this end, in step 10 we guess a state λ' and a stack γ' that the PDA moves into after looping, and in step 12 we check whether the PDA can move from state λ with stack γ to state λ' with stack γ' using a role chain compatible with the constant is moving into as given by the stationary PDA $\mathsf{spda}(\lambda, \gamma, \lambda', \gamma', o_{R,A})$. Because not all PDAs are required to loop, the stationary PDA $\mathsf{spda}(\lambda, \gamma, \lambda', \gamma', o_{R,A})$ can accept the empty word in case its start state and stack coincide with its final state and stack. Algorithm 3 thus checks loops only *after* each move, which is why step 17 in Algorithm 2 is necessary. Steps 2–5 take into account that each of the PDAs is nondeterministic and so it can initially make several ε-transitions; note that an explicit check for ε-transitions is required only initially since step 12 allows for possible ε-transitions after each move along a direct edge. Finally, we ensure termination of Algorithm 3 by observing that, since the stack of each PDA in $L(v, v')$ is bounded, the number of current configurations of each of the PDAs is exponential, and

so the number of distinct tuples of the current PDAs configurations is exponential as well; hence, the algorithm repeats computations after at most exponentially many steps. We thus obtain a nondeterministic decision procedure running in polynomial space by using a binary counter to stop the computation after all distinct configurations have been explored.

Instead we check property (b) in function check_{RI} directly by exploiting the well-known correspondence between PDAs, CFGs, and datalog programs [1]. In particular, for each root node v_r of $\mathcal{S}$ and each pair $\langle s, \mathcal{P} \rangle \in L(v_r)$, we construct in polynomial time the normalised CFG $G_{\mathcal{P}}$ that generates $\mathcal{L}(\mathcal{P})$ (see Proposition 2.1); then, we encode $G_{\mathcal{P}}$ using a datalog program $\mathsf{D}_{\mathcal{P}}$ such that, for all datalog constants u and u', we have that $\mathsf{D}_{\mathcal{K}} \cup \mathsf{D}_{\mathcal{P}} \models S_{\mathcal{L}}(u, u')$ if and only if a role chain $\rho \in \mathcal{L}(\mathcal{P})$ exists labelling a path from u to u' in J. Because $G_{\mathcal{P}}$ is normalised, each rule in $\mathsf{D}_{\mathcal{P}}$ contains a fixed number of variables, so the entailment check can be implemented to run in polynomial time.

9.2 Formalisation

We now formalise the intuitions from the previous section. Towards this goal, we fix a consistent $\mathcal{ELRO}^{+}_{\bot}$ KB $\mathcal{K} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$ with a regular RBox $\mathcal{R}$, a conjunctive query q', and a candidate answer τ to q' over $\mathsf{D}_{\mathcal{K}}$.

We use several notions that have been introduced in previous chapters. Let the canonical representative $\tau(t)_{\approx}$ for a term $t \in \mathsf{term}(q')$, and sets $\mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$ and $\mathsf{r\text{-}ind}_{\mathsf{D}_{\mathcal{K}}}$ be as specified in Definition 5.6 on page 57. Also, we use the notions of *good* and *aux-simple* atom, *connection graph*, *variable renaming*, and *skeleton* from Chapter 6. Moreover, we let function $\mathsf{isDSound}$ be as specified in Definition 6.5; and we let conjunctive queries q and $q_{\sim}$ be as specified in Definitions 6.1 and 6.3, respectively. Finally, we let $Q_{\mathcal{R}}$, $\Gamma_{\mathcal{R}}$, and $\delta_{\mathcal{R}}$ be as specified in Definition 8.1 on page 123, we let $\vdash$ be the derivation relation associated with $\delta_{\mathcal{R}}$, and we let $\mathsf{d}_{\mathcal{R}}$ be the depth of $\mathcal{R}$ as specified in Definition 8.4 on page 127.

We next generalise the notion of a PDA encoding the RBox $\mathcal{R}$ from Definition 8.1 by allowing arbitrary start and final states as well as arbitrary start and final stacks of size at most $\mathsf{d}_{\mathcal{R}}$; we will use these PDA to label the skeleton's edges. In addition, we associate each generalised PDA with a normalised context-free grammar that generates the PDA's

language; by Proposition 2.1, such grammar can be computed in polynomial time.

Definition 9.1. *The* generalised PDA *for an RBox $\mathcal{R}$, states $\lambda, \lambda' \in Q_{\mathcal{R}}$ and stack words $\gamma, \gamma' \in \Gamma_{\mathcal{R}}^*$ such that $|\gamma| \leq \mathsf{d}_{\mathcal{R}}$ and $|\gamma'| \leq \mathsf{d}_{\mathcal{R}}$, is defined as*

$$\mathsf{pda}(\lambda, \gamma, \lambda', \gamma') = \langle Q_{\mathcal{R}}, \mathsf{rol}_{\mathcal{R}}, \Gamma_{\mathcal{R}}, \delta_{\mathcal{R}}, \lambda, \gamma, \lambda', \gamma' \rangle.$$

Furthermore, let $G(\lambda, \gamma, \lambda', \gamma')$ be the normalised context-free grammar that generates the language of $\mathsf{pda}(\lambda, \gamma, \lambda', \gamma')$ and let $S_{\mathcal{L}}$ be the grammar's start symbol.

In Definition 9.2, we associate each generalised PDA $\mathsf{pda}(\lambda, \gamma, \lambda', \gamma')$ with a datalog program $\mathsf{rls}(\lambda, \gamma, \lambda', \gamma')$ that encodes $\mathcal{L}(\mathsf{pda}(\lambda, \gamma, \lambda', \gamma'))$. We will use the program $\mathsf{rls}(\lambda, \gamma, \lambda', \gamma')$ to efficiently check the constraints associated with the skeleton's roots. We obtain the datalog program by using the normalised context-free grammar $G(\lambda, \gamma, \lambda', \gamma')$ associated with $\mathsf{pda}(\lambda, \gamma, \lambda', \gamma')$ and by exploiting the well-known correspondence between datalog rules and production rules in context-free grammars [1]. Because the grammar $G(\lambda, \gamma, \lambda', \gamma')$ is normalised, each production rule $T \to T_1 \cdots T_n$ of the CFG is such that $n \leq 2$. Therefore, by Proposition 5.5, the set all of consequences of $\mathsf{D}_{\mathcal{K}} \cup \mathsf{rls}(\lambda, \gamma, \lambda', \gamma')$ can be computed in polynomial time.

Definition 9.2. *For states $\lambda, \lambda' \in Q_{\mathcal{R}}$ and stack words $\gamma, \gamma' \in \Gamma_{\mathcal{R}}$ such that $|\gamma| \leq \mathsf{d}_{\mathcal{R}}$ and $|\gamma'| \leq \mathsf{d}_{\mathcal{R}}$, the datalog program $\mathsf{rls}(\lambda, \gamma, \lambda', \gamma')$ is the smallest set that contains*

- *a rule $\top_c(x) \to S_{\mathcal{L}}(x, x)$ if $S_{\mathcal{L}} \to \epsilon$ is a production rule of $G(\lambda, \gamma, \lambda', \gamma')$, and*
- *a rule $\bigwedge_{i=1}^{n} T_i(x_{i-1}, x_i) \to T(x_0, x_n)$ for each production rule $T \to T_1 \cdots T_n$ with $n > 0$ of $G(\lambda, \gamma, \lambda', \gamma')$.*

For an auxiliary constant $o_{R,A}$, we next define the stationary PDA $\mathsf{spda}(\lambda, \gamma, \lambda', \gamma', o_{R,A})$ whose language contains all role chains $\rho \in \mathcal{L}(\mathsf{pda}(\lambda, \gamma, \lambda', \gamma'))$ such that $\mathsf{D}_{\mathcal{K}} \models \mathbb{S}_T(o_{R,A})$ for each role T occurring in ρ. By Proposition 5.5, we can compute PDA $\mathsf{spda}(\lambda, \gamma, \lambda', \gamma', o_{R,A})$ in time polynomial in $|\mathcal{K}|$.

Definition 9.3. *The* stationary PDA *for an auxiliary constant $o_{R,A}$, an RBox $\mathcal{R}$, and states $\lambda, \lambda' \in Q_{\mathcal{R}}$ and stack words $\gamma, \gamma' \in \Gamma_{\mathcal{R}}^*$ such that $|\gamma| \leq \mathsf{d}_{\mathcal{R}}$ and $|\gamma'| \leq \mathsf{d}_{\mathcal{R}}$, is defined as*

$$\mathsf{spda}(\lambda, \gamma, \lambda', \gamma', o_{R,A}) = \langle Q_{\mathcal{R}}, \mathsf{rol}_{\mathcal{R}}, \Gamma_{\mathcal{R}}, \delta_{R,A}, \lambda, \gamma, \lambda', \gamma' \rangle,$$

where $\delta_{R,A}$ is the smallest transition function that satisfies the following two properties for all states $\lambda_i, \lambda_i' \in Q_{\mathcal{R}}$, each symbol $X_i \in \Gamma_{\mathcal{R}}$, each role $T_i \in \mathsf{rol}_{\mathcal{R}}$, and each word $\gamma_i' \in \Gamma_{\mathcal{R}}^*$.

- *$\langle \lambda_i', \gamma_i' \rangle \in \delta_{\mathcal{R}}(\lambda_i, \varepsilon, X_i)$ implies that $\langle \lambda_i', \gamma_i' \rangle \in \delta_{R,A}(\lambda_i, \varepsilon, X_i)$.*
- *$\langle \lambda_i', \gamma_i' \rangle \in \delta_{\mathcal{R}}(\lambda_i, T_i, X_i)$ and $\mathsf{D}_{\mathcal{K}} \models \mathbb{S}_{T_i}(o_{R,A})$ imply that $\langle \lambda_i', \gamma_i' \rangle \in \delta_{R,A}(\lambda_i, T_i, X_i)$.*

Algorithm 3 shows the auxiliary procedure exist_{RI} that checks whether one can satisfy property (a) on page 149 for each skeleton edge $\langle v, v' \rangle \in \mathcal{S}$ that is labelled by a (possibly empty) set $\{\mathcal{P}_j = \mathsf{pda}(\lambda_j, \gamma_j, \lambda_j', \gamma_j') \mid 1 \leq j \leq m\}$ of generalised PDAs.

Finally, function check_{RI} in Definition 9.4 uses the auxiliary procedure exist_{RI} and the datalog program associated with each generalised PDA to determine whether one can satisfy both properties (a) and (b) for each vertex $v \in \mathcal{V}$.

Definition 9.4. *Given a skeleton vertex v, a constant u, and a function L that maps each skeleton edge to a finite set of generalised PDAs and each skeleton vertex to a finite set of pairs of the form $\langle s, \mathsf{pda}(\lambda, \gamma, \lambda', \gamma') \rangle$ consisting of a skeleton vertex and a generalised PDA, function $\mathsf{check}_{RI}(v, u, L)$ returns t if and only if the following two conditions hold.*

(a) Function $\mathsf{exist}_{RI}(u, \tau(v'), L(v, v'))$ returns t for each skeleton edge $\langle v, v' \rangle \in \mathcal{S}$.

(b) $\mathsf{D}_{\mathcal{K}} \cup \mathsf{rls}(\lambda, \gamma, \lambda', \gamma') \models S_{\mathcal{L}}(\tau(s), u)$ for each pair $\langle s, \mathsf{pda}(\lambda, \gamma, \lambda', \gamma') \rangle \in L(v)$.

The soundness of the candidate answer τ to q' over $\mathsf{D}_{\mathcal{K}}$ can be checked using the nondeterministic procedure $\mathsf{isSound}_{RI}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ from Algorithm 2. The next theorem shows how to use function $\mathsf{isSound}_{RI}$ to compute the certain answers to q' over $\mathcal{K}$. The proof of this result is given in Section 9.3.

Theorem 9.5. *For each substitution π, $\mathcal{K} \models_{\mathcal{DL}} \pi(q')$ if and only if a candidate answer τ to q' over $\mathsf{D}_{\mathcal{K}}$ exists such that $\tau|_{\vec{x}} = \pi$ and the following conditions hold:*

Algorithm 2: $\mathsf{isSound}_{RI}(q, \mathsf{D}_\mathcal{K}, \tau)$

if $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau) = \mathsf{f}$ **then return** f
return t **if** each $R(s,t) \in q_\sim$ is good or aux-simple
guess a variable renaming σ for q and τ
guess a skeleton $\mathcal{S} = \langle \mathcal{V}, \mathcal{E} \rangle$ for q, σ, and τ
foreach $v \in \mathcal{V}$, **let** $L(v) = \emptyset$; **foreach** $\langle v, v' \rangle \in \mathcal{E}$, **let** $L(v, v') = \emptyset$
foreach *aux-simple atom* $R(s,t) \in \sigma(q_\sim)$, **add** $\mathsf{pda}(i_R, \bot, f_R, \bot)$ to $L(s,t)$
foreach *neither good nor aux-simple atom* $R(s,t)$ *in* $\sigma(q_\sim)$ **do**
    let v_t be the unique root such that t is reachable from v_t in $\mathcal{S}$
    guess $v_0 \in \{s, v_t\}$ such that t is reachable from v_0 in $\mathcal{S}$
    let $v_0, \ldots, v_n$ be the unique path in $\mathcal{S}$ with $v_n = t$
    guess states $\lambda_0, \ldots, \lambda_n$ in $Q_\mathcal{R}$ with $\lambda_n = f_R$
    guess words $\gamma_0, \ldots \gamma_n$ in $\Gamma_\mathcal{R}^*$ with $\gamma_n = \bot$ and $|\gamma_i| \leq \mathsf{d}_\mathcal{R}$ $\forall i \in [0..n]$
    foreach $i \in [1..n]$ **add** $\mathsf{pda}(\lambda_{i-1}, \gamma_{i-1}, \lambda_i, \gamma_i)$ to $L(v_{i-1}, v_i)$
    if v_0 *is a root of* $\mathcal{S}$ **then**
        add $\langle s, \mathsf{pda}(i_R, \bot, \lambda_0, \gamma_0) \rangle$ to $L(v_0)$
    else
        return f **if** $\mathcal{L}(\mathsf{spda}(i_R, \bot, \lambda_0, \gamma_0, \tau(v_0))) = \emptyset$
foreach $v \in \mathcal{V}$ **do**
    if $v \in \Psi_q$ **then**
        return f **if** no individual $a \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ exists such that $\mathsf{check}_{RI}(v, a, L) = \mathsf{t}$
    else
        return f **if** $\mathsf{check}_{RI}(v, \tau(v), L) = \mathsf{f}$
return t

Algorithm 3: $\mathsf{exist}_{RI}(u, u', \{\mathcal{P}_j = \mathsf{pda}(\lambda_j, \gamma_j, \lambda'_j, \gamma'_j) \mid 1 \leq j \leq m\})$

let $const = u$ and **let** $M = (1 + |\mathsf{aux}_{\mathsf{D}_\mathcal{K}}|) \cdot |Q_\mathcal{R}|^m \cdot (|\Gamma_\mathcal{R}|^{1+\mathsf{d}_\mathcal{R}})^m$
for $j = 1$ **to** m **do**
    guess a state $\lambda \in Q_\mathcal{R}$ and a word $\gamma \in \Gamma_\mathcal{R}^*$ such that $|\gamma| \leq \mathsf{d}_\mathcal{R}$
    if $\epsilon \notin \mathcal{L}_{\mathsf{d}_\mathcal{R}}(\mathsf{pda}(\lambda_j, \gamma_j, \lambda, \gamma))$ **then return** f
    set $\mathsf{state}[j] = \lambda$ and **set** $\mathsf{stack}[j] = \gamma$
guess $k \in \mathbb{N}$ such that $1 \leq k \leq M$
for $r = 1$ **to** k **do**
    guess $o_{R,A} \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ such that $\mathsf{D}_\mathcal{K} \models \mathbb{D}_R(const, o_{R,A})$
    for $j = 1$ **to** m **do**
        guess $\{\lambda, \lambda'\} \subseteq Q_\mathcal{R}$ and $\{\gamma, \gamma'\} \subseteq \Gamma_\mathcal{R}^*$ with $|\gamma| \leq \mathsf{d}_\mathcal{R}$ and $|\gamma'| \leq \mathsf{d}_\mathcal{R}$
        if $\langle \mathsf{state}[j], R, \mathsf{stack}[j] \rangle \not\models \langle \lambda, \epsilon, \gamma \rangle$ **then return** f
        if $\mathcal{L}_{\mathsf{d}_\mathcal{R}}(\mathsf{spda}(\lambda, \gamma, \lambda', \gamma', o_{R,A})) = \emptyset$ **then return** f
        set $\mathsf{state}[j] = \lambda'$ and **set** $\mathsf{stack}[j] = \gamma'$
    set $const = o_{R,A}$
if $const \neq u'$ **then return** f
if $j \in [1..m]$ *exists such that* $\mathsf{state}[j] \neq \lambda'_j$ *or* $\mathsf{stack}[j] \neq \gamma'_j$ **then return** f
return t

(1) for each $x \in \vec{x}$, $\tau(x) \in N_{\mathfrak{I}}$, *and*

(2) a nondeterministic computation exists such that $\mathsf{isSound}_{RI}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ *returns* t.

Finally, we determine the complexity of function $\mathsf{isSound}_{RI}$. Towards this goal, we next determine the complexity of function exist_{RI}.

Lemma 9.6. *Function* $\mathsf{exist}_{RI}(u, u', \{\mathcal{P}_j = \mathsf{pda}(\lambda_j, \gamma_j, \lambda'_j, \gamma'_j) \mid 1 \leq j \leq m\})$ *in Algorithm 3 can be implemented so that it uses space polynomial in* $m \cdot |\mathcal{K}|$ *and, if the RBox* $\mathcal{R}$ *is fixed, it runs in time polynomial in* $|\mathcal{T}| + |\mathcal{A}|$.

Proof. Consider arbitrary u, u', and $\mathcal{P}_j = \mathsf{pda}(\lambda_j, \gamma_j, \lambda'_j, \gamma'_j)$ as stated above; let M be as in Algorithm 3. By the definition of generalised PDAs, we have $|\gamma_j| \leq \mathsf{d}_{\mathcal{R}}$ and $|\gamma'_j| \leq \mathsf{d}_{\mathcal{R}}$ for each $j \in [1..m]$.

By Proposition 2.2, using polynomial time one can compute PDAs that recognise the languages $\mathcal{L}_{\mathsf{d}_{\mathcal{R}}}(\mathsf{pda}(\lambda_j, \gamma_j, \lambda, \gamma))$ and $\mathcal{L}_{\mathsf{d}_{\mathcal{R}}}(\mathsf{spda}(\lambda, \gamma, \lambda', \gamma', o_{R,A}))$ in lines 4 and 12; so the checks in lines 4 and 12 can be implemented so that they use time (and thus space) polynomial in $|\mathcal{K}|$ [50].

For the space usage of Algorithm 3, please observe that the function stores the following information at each computation step:

(a) two arrays state and stack of length m such that $\mathsf{state}[j] \in Q_{\mathcal{R}}$, $\mathsf{stack}[j] \in \Gamma^*_{\mathcal{R}}$, and $|\mathsf{stack}[j]| \leq \mathsf{d}_{\mathcal{R}}$ for each $j \in [1..m]$,

(b) a generalised PDA in line 4,

(c) a stationary PDA in line 12,

(d) a constant $const \in \{u\} \cup \mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$ in line 1,

(e) a binary counter k such that $1 \leq k \leq M$, and

(f) the depth $\mathsf{d}_{\mathcal{R}}$ of $\mathcal{R}$, an auxiliary constant $o_{R,A}$, and role R.

By the definition of $\mathsf{d}_{\mathcal{R}}$, we have that $\mathsf{d}_{\mathcal{R}}$ is linearly bounded by the number of axioms occurring in $\mathcal{R}$; hence, we need at most $O(m \cdot |\mathcal{R}|)$ space to store the two arrays. Moreover, we need at most $O(m \cdot |\mathcal{K}|)$ space to store the counter k using binary encoding. By Definition

9.3 of stationary PDA, the size of $\mathsf{spda}(\lambda, \gamma, \lambda', \gamma', o_{R,A})$ is polynomial in $|\mathcal{R}|$; by Definition 8.1, the size of $\mathsf{pda}(\lambda, \gamma, \lambda', \gamma')$ is polynomial in $|\mathcal{R}|$. Overall, the space needed to store the required information is polynomial in $m \cdot |\mathcal{K}|$. By Proposition 5.5 we can realise the check in line 8 in polynomial time. Thus, exist_{RI} can be implemented so that it uses space polynomial in $m \cdot |\mathcal{K}|$.

Next, assume that the RBox $\mathcal{R}$ is fixed. Then $\mathsf{d}_{\mathcal{R}}$, $Q_{\mathcal{R}}$, $\mathsf{rol}_{\mathcal{R}}$, and $\Gamma_{\mathcal{R}}$ are all fixed as well; moreover, m is bounded by the size of $\mathcal{R}$ and so it is fixed, and M is linear in the size of $\mathcal{T}$ and $\mathcal{A}$. Thus, the number of alternatives in the nondeterministic step in step 3 of Algorithm 3 is fixed, so lines 1–5 require time polynomial in $|\mathcal{T}| + |\mathcal{A}|$. Furthermore, instead of guessing k using the nondeterministic step in line 6, we can repeat lines 7–14 for each $k \in [1..M]$, which requires a linear number of iterations. To show that lines 7–14 can also be implemented to run in polynomial time, we first define three sets which can be used to perform the checks in lines 8, 11, and 12.

$$\{\langle R, c, o_{R,A}\rangle \mid \text{for } R \in \mathsf{rol}_{\mathcal{R}}, c \in \{u\} \cup \mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}, o_{R,A} \in \mathsf{aux}_{\mathsf{D}_{\mathcal{K}}} \text{ with } \mathsf{D}_{\mathcal{K}} \models \mathbb{D}_R(c, o_{R,A})\} \quad (9.1)$$

$$\{\langle R, \mathsf{pda}(\lambda, \gamma, \lambda', \gamma')\rangle \mid \text{for } R \in \mathsf{rol}_{\mathcal{R}} \text{ with } \langle \lambda, R, \gamma\rangle \vdash \langle \lambda', \epsilon, \gamma'\rangle\} \quad (9.2)$$

$$\{\mathsf{spda}(\lambda, \gamma, \lambda', \gamma', o_{R,A}) \mid \mathcal{L}_{\mathsf{d}_{\mathcal{R}}}(\mathsf{spda}(\lambda, \gamma, \lambda', \gamma', o_{R,A})) \neq \emptyset\} \quad (9.3)$$

Given that $\mathcal{R}$ is fixed, these sets can be computed in time polynomial in the size of $\mathcal{T}$ and $\mathcal{A}$. We next show that we can implement the for-loop in lines 7–14 to use space logarithmic in the size of $\mathcal{T}$, $\mathcal{A}$, and of the sets in equations (9.1)–(9.3). For the space usage in lines 7–14, at each computation step of the for-loop we store the information from points (a)–(f) above. Since $\mathcal{R}$ and m are fixed, however, points (a)–(c) require constant space. Furthermore, the checks in lines 8, 11, and 12 can be performed by a lookup in sets (9.1)–(9.3); by storing these sets using a suitable binary encoding and by using a binary index into the sets, this check can be implemented using logarithmic space. Finally, as $\mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$ and M are linear in the size of $\mathcal{T}$ and $\mathcal{A}$, we can store counter k using a binary encoding, so the overall space the function needs to store is logarithmic in $|\mathcal{T}| + |\mathcal{A}|$, and the size of the sets (9.1)–(9.3). Thus, lines 7–14 require nondeterministic logarithmic space, and it is well known that this

implies that lines 7–14 can be implemented to run in polynomial time. Finally, lines 15–17 clearly require polynomial time. Therefore, function exist_{RI} can be implemented so that it runs in time polynomial in $|\mathcal{T}| + |\mathcal{A}|$ for fixed $\mathcal{R}$. □

We are now ready to establish the complexity of function $\mathsf{isSound}_{RI}$.

Theorem 9.7. *Function* $\mathsf{isSound}_{RI}(q, \mathsf{D}_\mathcal{K}, \tau)$ *can be implemented so that*

(1) it uses space polynomial in the input size,

(2) if each binary atom in $q_\sim$ *is either good or aux-simple, it runs in polynomial time in the input size,*

(3) if the RBox $\mathcal{R}$ *is fixed, it runs in nondeterministic polynomial time in the size of the TBox* $\mathcal{T}$*, the ABox* $\mathcal{A}$*, the query* q*, and the substitution* τ*, and*

(4) if the RBox $\mathcal{R}$ *and the query* q *are fixed, it runs in polynomial time in the size of the TBox* $\mathcal{T}$ *and the ABox* $\mathcal{A}$*.*

Proof. We can check whether the language of a PDA is empty in time polynomial in $|\mathcal{K}|$ [50]. Moreover, the number of variables occurring in each rule in the program $\mathsf{rls}(i_R, \bot, \lambda_0, \gamma_0)$ is fixed, and so the set of all consequences of $\mathsf{D}_\mathcal{K} \cup \mathsf{rls}(i_R, \bot, \lambda_0, \gamma_0)$ can be computed in time polynomial in $|\mathcal{K}|$ [24]. Finally, by Lemma 6.11, function $\mathsf{isDSound}$ runs in polynomial time in the input size; therefore, the checks in lines 1 and 2, 17, and the check (b) in Definition 9.4 of function check_{RI} all require time (and therefore space) polynomial in the input size.

Property (1) and (2). Please note that the function $\mathsf{isSound}_{RI}$ as specified in Algorithm 2 stores the following information at each computation step:

- the CQ q, datalog program $\mathsf{D}_\mathcal{K}$, and substitution τ;
- a variable renaming σ and CQ $\sigma(q_\sim)$;
- a skeleton $\mathcal{S} = \langle \mathcal{V}, \mathcal{E} \rangle$ and the connection graph $\mathsf{cg} = \langle V, E_s, E_t \rangle$ from Definition 6.4 which we use to guess $\mathcal{S}$ more efficiently;
- a binary atom $R(s, t)$;

- a path $v_0, \ldots, v_n$ in $\mathcal{S}$, a sequence of states $\lambda_0, \ldots, \lambda_n$ in $Q_{\mathcal{R}}$, and a sequence of words $\gamma_0, \ldots, \gamma_n$ in $\Gamma_{\mathcal{R}}^*$ such that $|\gamma_i| \leq \mathsf{d}_{\mathcal{R}}$ for each $i \in [0..n]$;
- a datalog program $\mathsf{rls}(i_R, \bot, \lambda_0, \gamma_0)$;
- a function L mapping each edge $\langle v, v' \rangle \in E$ to a set of generalised PDA and each vertex $v \in \mathcal{V}$ to a set of pairs consisting of a term and a generalised PDA; and,
- a stationary PDA $\mathsf{spda}(i_R, \bot, \lambda_0, \gamma_0, \tau(v_0))$.

By Definition 6.7 of skeleton and Definition 6.4 of connection graph, we need space polynomial in the size of q and $\mathcal{K}$ to store $\mathcal{S}$ and cg. Moreover, the length of the longest path in $\mathcal{S}$ is given by the number of variables occurring in $\sigma(q_\sim)$, so we can store the sequences of vertices, states, and words in space polynomial in $|q|$ and $|\mathcal{K}|$ as well. Also, each set $L(v, v')$ and each set $L(v)$ contain at most m PDAs and m pairs, respectively, where m is the number of binary atoms occurring in $\sigma(q_\sim)$. Also, by Lemma 9.6, check (a) in Definition 9.4 of function check_{RI} can be implemented so that it uses space polynomial in the input size; thus, also $\mathsf{isSound}_{RI}$ can be implemented so that it uses space polynomial in the input size and property (1) holds. Because function $\mathsf{isDSound}$ runs in polynomial time, property property (2) immediately holds as well.

Property (3). Assume that the RBox $\mathcal{R}$ is fixed. By Lemma 9.6, for a fixed RBox $\mathcal{R}$, check (a) in Definition 9.4 of function check_{RI} can be implemented to run in time polynomial in $|\mathcal{T}|+|\mathcal{A}|$; hence, for $\mathcal{R}$ fixed, function check_{RI} runs in time polynomial in $|\mathcal{T}|+|\mathcal{A}|$. Clearly, all other steps in Algorithm 2 can be implemented to run in nondeterministic polynomial time in the size of TBox $\mathcal{T}$, ABox $\mathcal{A}$, and CQ q. Thus, for a fixed RBox $\mathcal{R}$, function $\mathsf{isSound}_{RI}$ can be implemented so that it runs in nondeterministic polynomial time in the size of TBox $\mathcal{T}$, ABox $\mathcal{A}$, and CQ q.

Property (4). Assume that the RBox $\mathcal{R}$ and the query q are fixed. Then $\mathsf{d}_{\mathcal{R}}$, $Q_{\mathcal{R}}$, $\mathsf{rol}_{\mathcal{R}}$, and $\Gamma_{\mathcal{R}}$ are all fixed as well. Given that the number of variables occurring in q is fixed, the number of guessing steps required in lines 3 and 4 is fixed; also, the number of alternatives for these steps is linear in $|\mathcal{T}| + |\mathcal{A}|$. Thus, lines 3 and 4 require polynomial time. Furthermore, the maximum number of iterations of the for-loop in lines 7–17 is fixed

and the length of the longest path in $\mathcal{S}$ is fixed. Thus, the number of guessing steps in lines 11 and 12 is also fixed. In addition, the number of alternatives for the guessing steps in lines 11 and 12 is fixed as well. Therefore, lines 7–17 require time polynomial in $|\mathcal{T}| + |\mathcal{A}|$. Finally, since the query is fixed, the maximum number of iterations of the for-loop in lines 18–22 is also fixed and so, by Lemma 9.6, the for-loop requires time polynomial in $\mathcal{T}$ and $\mathcal{A}$. Therefore, $\mathsf{isSound}_{RI}$ can be implemented so that it runs in time polynomial in $|\mathcal{T}| + |\mathcal{A}|$ for fixed $\mathcal{R}$ and q. □

By Propositions 5.5 and 5.7, we can check whether an $\mathcal{ELRO}^+_\bot$ KB is inconsistent using polynomial time; hence, by Theorem 6.10 and Savitch's theorem, checking whether π is a certain answer to q over $\mathcal{K}$ is in PSPACE in combined complexity (i.e., when the ABox, the RBox, the TBox, and the query are all part of the input), in NP if the RBox is fixed (i.e., when the ABox, the TBox, and the query are part of the input), and in PTIME in data complexity (i.e., when the TBox, the RBox, and the query are fixed). Calvanese et al. [13] showed that instance checking over $\mathcal{EL}$ knowledge bases is PTIME-hard in data complexity; when the query is not fixed, CQ answering is NP-hard already over relational databases [18]; and Krötzsch et al. [66] showed that the problem is PSPACE-hard in combined complexity. In addition, in Chapter 10, we shall improve the PSPACE lower bound by Krötzsch et al. [66] and show that CQ answering over $\mathcal{ELRO}^+_\bot$ knowledge bases is PSPACE-hard already in the restricted setting where the query, the TBox, and the ABox are all fixed and just the RBox varies; thus, showing that the problem is PSPACE-hard even in knowledge base complexity. We next summarise these results; the proof of the PSPACE-hardness in knowledge base complexity is given in Theorem 10.7 on page 189.

Theorem 9.8. *For π a substitution, checking whether $\mathcal{K} \models_{\mathcal{DL}} \pi(q')$ is*

- PTIME*-complete in data complexity,*
- NP*-complete, if the RBox $\mathcal{R}$ is fixed, and*
- PSPACE*-complete in combined and KB complexities.*

9.3 Proof of Correctness

In the rest of this section, let $\Xi_{\mathcal{K}}$ be the rule base for $\mathcal{K}$; furthermore, let I and J be universal interpretations of $\Xi_{\mathcal{K}}$ and $\mathsf{D}_{\mathcal{K}}$, respectively, and let ι be the mapping from the terms occurring in I to the terms occurring in J that we specified in Section 5.3.

We start by proving the correctness of function exist_{RI}; then we will prove that datalog program $\mathsf{rls}(\lambda, \gamma, \lambda', \gamma')$ correctly encodes a generalised PDA $\mathsf{pda}(\lambda, \gamma, \lambda', \gamma')$; and finally, we will prove that function $\mathsf{isSound}_{RI}$ is sound and complete.

9.3.1 Correctness of exist_{RI}

The following lemma proves the correctness of function exist_{RI} from Algorithm 3.

Lemma 9.9. *Function* $\mathsf{exist}_{RI}(u, u', \{\mathcal{P}_j = \mathsf{pda}(\lambda_j, \gamma_j, \lambda'_j, \gamma'_j) \mid 1 \leq j \leq m\})$ *returns* t *if and only if a natural number* $k \geq 1$*, auxiliary constants* $u_1, \ldots, u_k$ *in* $\mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$ *with* $u_k = u'$*, a set of role chains* $\{\chi_{j,i} \mid j \in [1..m]$ *and* $i \in [1..k]\}$*, and roles* $R_1, \ldots, R_k$ *and concepts* $A_1, \ldots, A_k$ *exist such that the following conditions hold for each* $j \in [1..m]$ *and each* $i \in [1..k]$*.*

(1) $u_0 = u$*,* $u_i = o_{R_i, A_i}$*, and* $\mathsf{D}_{\mathcal{K}} \models \mathbb{D}_{R_i}(u_{i-1}, u_i)$*.*

(2) For each role T *occurring in* $\chi_{j,i}$*, we have that* $\mathsf{D}_{\mathcal{K}} \models \mathbb{S}_T(u_i)$*.*

(3) $R_1 \cdot \chi_{j,1} \cdots R_k \cdot \chi_{j,k} \in \mathcal{L}_{\mathsf{d}_{\mathcal{R}}}(\mathcal{P}_j)$*.*

Proof. Consider u, u', and $\mathcal{P}_j = \mathsf{pda}(\lambda_j, \gamma_j, \lambda'_j, \gamma'_j)$ as stated in the lemma; and let $u_0 = u$.

($\Rightarrow$) Assume that there is a nondeterministic computation of exist_{RI} such that the function returns t. Let $k \in \mathbb{N}$ be as guessed in line 6; we show that the for-loop in lines 7–14 satisfies the following invariant: after each iteration r, there exist constants $u_1, \ldots, u_r$ in $\mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$ with $u_r = const$, roles $R_1, \ldots, R_r$, concepts $A_1, \ldots, A_r$, and role chains $\chi_{j,1}, \ldots, \chi_{j,r}$ for each $j \in [1..m]$ such that the following holds for each $i \in [1..r]$ and $j \in [1..m]$.

(a) $u_i = o_{R_i, A_i}$ and $\mathsf{D}_{\mathcal{K}} \models \mathbb{D}_{R_i}(u_{i-1}, u_i)$.

(b) For each role T occurring in $\chi_{j,i}$, we have that $\mathsf{D}_{\mathcal{K}} \models \mathbb{S}_T(u_i)$.

(c) $R_1 \cdot \chi_{j,1} \cdots R_r \cdot \chi_{j,r} \in \mathcal{L}_{\mathsf{d}_{\mathcal{R}}}(\mathsf{pda}(\lambda_j, \gamma_j, \mathsf{state}[j], \mathsf{stack}[j]))$.

$$
\begin{array}{rcll}
 & & \langle \lambda_j,\ R_1 \cdot \chi_{j,1} \cdots R_n \cdot \chi_{j,n},\ \gamma_j \rangle & \vdash^* \\
\langle \lambda_{j,0},\ R_1 \cdot \chi_{j,1} \cdots R_n \cdot \chi_{j,n},\ \gamma_{j,0} \rangle & \vdash & \langle \lambda'_{j,1},\ \chi_{j,1} \cdot R_2 \cdots R_n \cdot \chi_{j,n},\ \gamma'_{j,1} \rangle & \vdash^* \\
\langle \lambda_{j,1},\ R_2 \cdot \chi_{j,2} \cdots R_n \cdot \chi_{j,n},\ \gamma_{j,1} \rangle & \vdash & \dots & \vdash^* \\
\langle \lambda_{j,i-1},\ R_i \cdot \chi_{j,i} \cdots R_n \cdot \chi_{j,n},\ \gamma_{j,i-1} \rangle & \vdash & \langle \lambda'_{j,i},\ \chi_{j,i} \cdot R_{i+1} \cdots R_n \cdot \chi_{j,n},\ \gamma'_{j,i} \rangle & \vdash^* \\
\langle \lambda_{j,i},\ R_{i+1} \cdot \chi_{j,i} \cdots R_n \cdot \chi_{j,n},\ \gamma_{j,i} \rangle & \vdash & \dots & \vdash^* \\
\langle \lambda_{j,n-1},\ R_n \cdot \chi_{j,n},\ \gamma_{j,n-1} \rangle & \vdash & \langle \lambda'_{j,n},\ \chi_{j,n},\ \gamma'_{j,n} \rangle & \vdash^* \\
\langle \lambda_{j,n},\ \epsilon,\ \gamma_{j,n} \rangle & & &
\end{array}
$$

Figure 9.4: The form of derivations of $\mathcal{P}_j$ for $R_1 \cdot \chi_{j,1} \cdots R_n \cdot \chi_{j,n}$

Base case. Before the first iteration of the loop (i.e., after lines 1–5 and for $r = 0$), for each $j \in [1..m]$, we have that $const = u = u_0$ and $\epsilon \in \mathcal{L}_{\mathsf{d}_\mathcal{R}}(\mathsf{pda}(\lambda_j, \gamma_j, \mathsf{state}[j], \mathsf{stack}[j]))$, so properties (a)–(c) clearly hold.

Inductive step. Consider an arbitrary iteration $r \in [1..k-1]$ and assume that properties (a)–(c) hold at the end of iteration r; we show that the same is true after iteration $r+1$. By the inductive hypothesis, there exist constants $u_1, \dots, u_r$ in $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ with $u_r = const$, roles $R_1, \dots, R_r$, concepts $A_1, \dots, A_r$, and role chains $\chi_{j,1}, \dots, \chi_{j,r}$ for each $j \in [1..m]$ that satisfy properties (a)–(c). Let constant $u_{r+1} = o_{R_{r+1}, A_{r+1}}$ be as guessed in line 8. Therefore, we have that $u_{r+1} \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ and $\mathsf{D}_\mathcal{K} \models \mathbb{D}_{R_i}(u_r, u_{r+1})$, as required for property (a). Furthermore, consider an arbitrary $j \in [1..m]$, let λ, λ', γ, and γ' be as guessed in line 10, and let $\mathsf{state}[j]$ and $\mathsf{stack}[j]$ be as at the end of iteration r; then, $\langle \mathsf{state}[j], R_{r+1}, \mathsf{stack}[j] \rangle \vdash \langle \lambda, \epsilon, \gamma \rangle$ due to line 11; also, due to the check in line 12, there exists a role chain $\chi_{j,r+1} \in \mathcal{L}_{\mathsf{d}_\mathcal{R}}(\mathsf{spda}(\lambda, \gamma, \lambda', \gamma', u_{r+1}))$. By Definition 9.3 of stationary PDA, for each role T occurring in $\chi_{j,r+1}$, we have $\mathsf{D}_\mathcal{K} \models \mathbb{S}_T(u_{r+1})$, as required for property (b), and $\chi_{j,r+1} \in \mathcal{L}_{\mathsf{d}_\mathcal{R}}(\mathsf{pda}(\lambda, \gamma, \lambda', \gamma'))$. Finally, after line 13, we have that

$$R_1 \cdot \chi_{j,1} \cdots R_r \cdot \chi_{j,r} \cdot R_{r+1} \cdot \chi_{j,r+1} \in \mathcal{L}_{\mathsf{d}_\mathcal{R}}(\mathsf{pda}(\lambda_j, \gamma_j, \mathsf{state}[j], \mathsf{stack}[j]))$$

so property (c) holds.

Line 15 ensures that $const = u'$; furthermore, lines 16–17 ensure that $\mathsf{state}[j] = \lambda'_j$ and $\mathsf{stack}[j] = \gamma'_j$ for each $j \in [1..m]$, so $R_1 \cdot \chi_{j,1} \cdots R_k \cdot \chi_{j,k} \in \mathcal{L}(\mathcal{P}_j)$. Thus, properties (1)–(3) of this lemma hold.

($\Leftarrow$) Let $u_1, \ldots, u_n$ be auxiliary constants in $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ with $u_n = u'$, let $R_1, \ldots, R_n$ be roles, let $A_1, \ldots, A_n$ be concepts with $u_i = o_{R_i, A_i}$ for each $i \in [1..n]$, and let $\chi_{j,i}$ be role chains satisfying properties (1)–(3) of this lemma. Figure 9.4 shows the form of each derivation for $R_1 \cdot \chi_{j,1} \cdots R_n \cdot \chi_{j,n}$ of PDA $\mathcal{P}_j$, where $\lambda_{j,n} = \lambda'_j$ and $\gamma_{j,n} = \gamma'_j$:

The initial transition moving $\mathcal{P}_j$ from state λ_j to state $\lambda_{j,0}$ is 'special' in the sense that it allows $\mathcal{P}_j$ to make an arbitrary number of ε-transitions; the rest of the derivation is regular and consists of reading R_i and $\chi_{j,i}$. Thus, $\lambda_{j,i-1}$ and $\lambda'_{j,i}$ are the states of $\mathcal{P}_j$ before and after, respectively, reading R_i, and $\gamma_{j,i-1}$ and $\gamma'_{j,i}$ are the respective stacks. By property (3) of this lemma, we have $|\gamma_{j,i}| \leq \mathsf{d}_\mathcal{R}$ and $|\gamma'_{j,i}| \leq \mathsf{d}_\mathcal{R}$.

Let $X_i = \langle u_i, \lambda_{1,i}, \gamma_{1,i}, \ldots, \lambda_{m,i}, \gamma_{m,i} \rangle$. Clearly, there are $|Q_\mathcal{R}|$ many different states in each PDA $\mathcal{P}_j$, and $1 + |\mathsf{aux}_{\mathsf{D}_\mathcal{K}}|$ different constants in $\{u\} \cup \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$; moreover, there are $\sum_{\ell=0}^{\mathsf{d}_\mathcal{R}} |\Gamma_\mathcal{R}|^\ell$ many different stacks of length at most $\mathsf{d}_\mathcal{R}$. Since $|\Gamma_\mathcal{R}| > 0$ and due to $\mathsf{d}_\mathcal{R} \geq 0$, we have that $\sum_{\ell=0}^{\mathsf{d}_\mathcal{R}} |\Gamma_\mathcal{R}|^\ell \leq |\Gamma_\mathcal{R}|^{1+\mathsf{d}_\mathcal{R}}$; consequently, there are at most M distinct such tuples. Therefore, for some $k \leq M$, we have that $X_k = X_n$. But then, $u_k = u'$; furthermore, for each $j \in [1..m]$, we have $\lambda_{j,k} = \lambda_{j,n} = \lambda'_j$ and $\gamma_{j,k} = \gamma_{j,n} = \gamma'_j$, so we have $R_1 \cdot \chi_{j,1} \cdots R_k \cdot \chi_{j,k} \in \mathcal{L}_{\mathsf{d}_\mathcal{R}}(\mathcal{P}_j)$.

We can now easily construct a nondeterministic computation of exist_{RI} as follows. In line 3, for each j we let $\lambda = \lambda_{j,0}$ and $\gamma = \gamma_{j,0}$; clearly, condition in line 4 is not satisfied. For each r in the for-loop in lines 7–14, we proceed as follows.

- In line 8 we let $o_{R,A} = u_r$; clearly, the condition is satisfied due to property (1).
- For each $j \in [1..m]$, let $\lambda = \lambda'_{j,r}$ and $\lambda' = \lambda_{j,r}$, and let $\gamma = \gamma'_{j,r}$ and $\gamma' = \gamma_{j,r}$; clearly, condition in line 11 is not satisfied due to the form of the derivation; condition in line 12 is also not satisfied due to property (2) and Definition 9.3 of stationary PDA.

Finally, conditions in lines 15 and 16 are not satisfied due to the way in which we chose k. Thus, function exist_{RI} returns true in line 17. □

9.3.2 Correctness of the Datalog Encoding of Generalised PDAs

The following result proves the correctness of the datalog encoding of the generalised PDA $\mathsf{pda}(\lambda, \gamma, \lambda', \gamma')$ from Definition 9.2. Recall that $S_\mathcal{L}$ is the start symbol of the context-free

grammar associated with $\mathsf{pda}(\lambda, \gamma, \lambda', \gamma')$.

Lemma 9.10. *For all states $\lambda, \lambda' \in Q_{\mathcal{R}}$, all stack words $\gamma, \gamma' \in \Gamma_{\mathcal{R}}^*$ with $|\gamma| \leq \mathsf{d}_{\mathcal{R}}$ and $|\gamma'| \leq \mathsf{d}_{\mathcal{R}}$, and all constants u and u' from $\mathsf{D}_{\mathcal{K}}$, $\mathsf{D}_{\mathcal{K}} \cup \mathsf{rls}(\lambda, \gamma, \lambda', \gamma') \models S_{\mathcal{L}}(u, u')$ if and only if roles $R_1, \ldots, R_n$ and constants $u_0, \ldots, u_n$ with $u_0 = u$ and $u_n = u'$ exist such that*

- $R_1 \cdots R_n \in \mathcal{L}(\mathsf{pda}(\lambda, \gamma, \lambda', \gamma'))$, *and*
- $\mathsf{D}_{\mathcal{K}} \models R_i(u_{i-1}, u_i)$ *for each* $i \in [1..n]$.

Proof Sketch. Consider states λ and λ' and stack words γ and γ' as stated in the Lemma. By Definition 9.1 of generalised PDA, the normalised context-free grammar $G(\lambda, \gamma, \lambda', \gamma')$ generates the language of $\mathsf{pda}(\lambda, \gamma, \lambda', \gamma')$. Let $\Rightarrow$ be the derivation relation for $G(\lambda, \gamma, \lambda', \gamma')$, and let V and Σ be the set of non-terminal and terminal symbols of the grammar, respectively. Because Abiteboul et al. [1, Chapter 12] already showed a similar correspondence between datalog programs and CFGs, we next provide a sketch of the proof.

($\Rightarrow$) Let K be a universal interpretation of $\mathsf{D}_{\mathcal{K}} \cup \mathsf{rls}(\lambda, \gamma, \lambda', \gamma')$. Furthermore, let K_0 be the chase instance that contains each ground atom in $\mathsf{D}_{\mathcal{K}} \cup \mathsf{rls}(\lambda, \gamma, \lambda', \gamma')$, and let $\langle K_1, r_1, \sigma_1 \rangle, \langle K_2, r_2, \sigma_2 \rangle, \ldots$ be the chase sequence for $\mathsf{D}_{\mathcal{K}} \cup \mathsf{rls}(\lambda, \gamma, \lambda', \gamma')$ used to construct K. By a straightforward induction on the length $n \in \mathbb{N}$ of the chase sequence one can show that, for all constants u and u' and each non-terminal symbol $X \in V$, we have that $X(u, u') \in \mathsf{inst}_{K_n}$ implies that roles $R_1, \ldots, R_n$ and constants $u_0, \ldots, u_n$ with $u_0 = u$ and $u_n = u'$ exist such that $X \Rightarrow^* R_1 \cdots R_n$ and $\mathsf{D}_{\mathcal{K}} \models R_i(u_{i-1}, u_i)$ for each $i \in [1..n]$.

($\Leftarrow$) Consider an arbitrary non-terminal symbol $X \in V$. In the following, for each word $\rho \in (V \cup \Sigma)^*$, we write $X \overset{0}{\Rightarrow} \rho$ if $\rho = X$; furthermore, for each $m \in \mathbb{N}^+$, we write $X \overset{m}{\Rightarrow} \rho$ if words $\rho_1, \ldots, \rho_m$ with $\rho_m = \rho$ exist in $(V \cup \Sigma)^*$ such that $X \Rightarrow \rho_1 \Rightarrow \cdots \Rightarrow \rho_m$. By a straightforward induction on $m \in \mathbb{N}$, one can show that, for all symbols $X_1, \ldots, X_n$ in $V \cup \Sigma$ and all constants $u_0, \ldots, u_n$ such that $X \overset{m}{\Rightarrow} X_1 \cdots X_n$ and $\mathsf{D}_{\mathcal{K}} \cup \mathsf{rls}(\lambda, \gamma, \lambda', \gamma') \models X_i(u_{i-1}, u_i)$ for each $i \in [1..n]$, we have that $\mathsf{D}_{\mathcal{K}} \cup \mathsf{rls}(\lambda, \gamma, \lambda', \gamma') \models X(u_0, u_n)$. □

9.3.3 Soundness

Assume that, for each $x \in \vec{x}$, we have $\tau(x) \in N_{\mathfrak{I}}$ and that a nondeterministic computation exists such that $\mathsf{isSound}_{RI}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ returns t; we show that $\mathcal{K} \models_{\mathcal{DL}} \tau|_{\vec{x}}(q')$. By the definition

of candidate answer and by Theorem 2.3, we have that $\|\tau(q')\|_J \subseteq \mathsf{inst}_J$. Furthermore, by the definition of CQ q, for each term $t \in \mathsf{term}(q)$, we have $\|\tau(t)\|_J = \tau(t)$; and so $\tau(q) \subseteq \mathsf{inst}_J$.

Lemma 6.20 shows that, if $\mathsf{isSound}_{RI}(q, \mathsf{D}_\mathcal{K}, \tau)$ returns t in line 2, then $\mathcal{K} \models_{\mathcal{DL}} \tau|_{\vec{x}}(q')$. Hence, in the following we consider only the case in which $\mathsf{isSound}_{RI}$ returns t in line 23.

Let $q_\sim$ be as specified in Definition 6.3, and let $\mathsf{cg} = \langle V, E_s, E_t \rangle$ be as specified in Definition 6.4. Furthermore, let variable renaming σ, skeleton $\mathcal{S} = \langle \mathcal{V}, \mathcal{E} \rangle$, function L, and set $L(v)$ for each $v \in \mathcal{V}$ be as determined by $\mathsf{isSound}_{RI}$.

In the rest of this proof, we construct a substitution ν that is congruent with τ (see Definition 6.16 on page 100) and satisfies the following properties:

(a) for all terms $s, t \in \mathsf{term}(q)$ such that $s \sim t$ or $\sigma(s) = t$, we have that $\nu(s) = \nu(t)$,

(b) for each edge $\langle v, v' \rangle \in \mathcal{E}$ and each PDA $\mathcal{P}_j \in L(v, v')$, there exists a non-empty role chain $\rho_j \in \mathcal{L}(\mathcal{P}_j)$ such that $\rho_j(\nu(v), \nu(v')) \in \mathsf{inst}_I$.

By the definition of a skeleton, graph $\mathcal{S}$ is a forest rooted in the individuals and in the placeholders occurring in $\mathcal{V}$. Then, we define ν by structural induction on the forest $\mathcal{S}$; later we show that $\nu(q) \subseteq \mathsf{inst}_I$.

Base case. Consider a root $v \in \mathcal{V}$. We distinguish two cases.

- $v \in \mathcal{V} \setminus \Psi_q$; therefore vertex v is a named individual in $\mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$. Given that each element in $\mathsf{rng}(\sigma)$ is a variable, no term $s \in \mathsf{term}(q)$ exists such that $\sigma(s) = v$. Then, for each term $s \in \mathsf{term}(q)$ with $s \sim v$, let $\nu(s) = v$. By condition (1) in Definition 6.5, we have $\tau(s) = \tau(v)$. Thus, properties (C) in Definition 6.16, and (a) are satisfied.

- $v \in \mathcal{V} \cap \Psi_q$. By the definitions of relation $\sim$ and renaming σ, no term $s \in \mathsf{term}(q)$ exists with $s \sim v$ or $\sigma(s) = v$. Then let $\nu(v)$ be an arbitrary individual $a \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ such that $\mathsf{check}_{RI}(v, a, L)$ return t. Since the condition in line 20 is not satisfied, such an individual exists; moreover, properties (C) of Definition 6.16, and (a) hold vacuously.

Inductive step. Consider $\langle v, v' \rangle \in \mathcal{E}$ such that $\nu(v)$ has been defined, but $\nu(v')$ has not. Let $u_0 = \tau(v)$ if $u_0 \notin \Psi_q$; otherwise, let $u_0 = \nu(v)$; furthermore, let $L(v, v') = \{\mathcal{P}_1, \ldots, \mathcal{P}_m\}$

the (possibly empty) set of PDAs associated with $\langle v, v' \rangle$. Because the checks in lines 20 and 22 fail and by the definition of procedure check_{RI}, function $\mathsf{exist}_{RI}(u_0, \tau(v'), L(v, v'))$ returns t. By Lemma 9.9, there exist auxiliary constants $u_1, \ldots, u_n$ in $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ with $u_n = \tau(v')$, roles $R_1, \ldots, R_k$, concepts $A_1, \ldots, A_k$, and a role chain $\rho_j = R_1 \cdot \chi_{j,1} \cdots R_k \cdot \chi_{j,k}$ for each $j \in [1..m]$ such that the following conditions hold for each $j \in [1..m]$ and each $i \in [1..k]$.

- $u_i = o_{R_i, A_i}$ and $\mathbb{D}_{R_i}(u_{i-1}, u_i) \in \mathsf{inst}_J$.
- For each role T occurring in $\chi_{j,i}$, we have that $\mathbb{S}_T(u_i) \in \mathsf{inst}_J$.
- $\rho_j \in \mathcal{L}_{\mathsf{d}_\mathcal{R}}(\mathcal{P}_j)$.

Let $w_0 = \nu(v)$. By property (d5) of Lemma 5.13, for each $i \in [1..n]$, a term w_i exists such that $\iota(w_i) = u_i$ and $R_i(w_{i-1}, w_i) \in \mathsf{inst}_I$; furthermore, by property (d2) of Lemma 5.13, for each $j \in [1..m]$ and each role T occurring in $\chi_{j,i}$, we have that $T(w_i, w_i) \in \mathsf{inst}_I$; and so $\rho_j(w_0, w_n) \in \mathsf{inst}_I$. Now, for each $s \in \mathsf{term}(q)$ with $s \sim v'$ or $\sigma(s) = v'$, let $\nu(s) = w_n$. Properties (a)–(b) clearly hold; property (C) is also satisfied, since $\sigma(s) = v'$ implies $\tau(s) = \tau(v')$, by construction of σ, and $s \sim v'$ implies $\tau(s) = \tau(v')$, by condition (1) in Definition 6.5.

Lemma 9.11. *Substitution ν is such that $\nu(q) \subseteq \mathsf{inst}_I$.*

Proof. We next show that $\nu(q) \subseteq \mathsf{inst}_I$ by considering the various atoms occurring in q.

Consider an atom $A(s)$ in q. By assumption, we have $A(\tau(s)) \in \mathsf{inst}_J$. Because ν is congruent with τ, by Lemma 6.17, we have that $A(\nu(s)) \in \mathsf{inst}_I$.

Consider an atom $R(s', t')$ in q. By the definition of $q_\sim$, terms s'' and t'' occur in $q_\sim$ such that $s' \sim s''$, $t' \sim t''$, and $R(s'', t'')$ is an atom in $q_\sim$. By condition (1) in Definition 6.5, we have $\tau(s') = \tau(s'')$ and $\tau(t') = \tau(t'')$. Therefore, $R(\tau(s''), \tau(t'')) \in \mathsf{inst}_J$. By the definition of $\sigma(q_\sim)$, terms s and t occur in $\sigma(q_\sim)$ such that $\sigma(s'') = s$, $\sigma(t'') = t$, and $R(s, t)$ is an atom in $\sigma(q_\sim)$. By the definition of variable renaming, we have $\tau(t'') = \tau(t)$ and $\tau(s'') = \tau(s)$; and so $R(\tau(s), \tau(t)) \in \mathsf{inst}_J$. By property (a), it suffices to show that $R(\nu(s), \nu(t)) \in I$. Towards this goal, we consider three cases.

$R(s, t)$ is good. As ν is congruent with τ, we have $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$ by Lemma 6.17.

$R(s, t)$ is aux-simple. By the definition of $\mathcal{E}$, we have $\langle s, t \rangle \in \mathcal{E}$; moreover, by step 6, we have $\mathsf{pda}(i_R, \bot, f_R, \bot) \in L(s, t)$. By property (b), a role chain $\rho_j \in \mathcal{L}(\mathsf{pda}(i_R, \bot, f_R, \bot))$

exists such that $\rho_j(\nu(s), \nu(t)) \in \mathsf{inst}_I$. By Theorem 8.3, we have that $\rho_j \in \mathcal{L}(R)$; since I is closed under $\Xi_{\mathcal{K}}$, R is simple, and $\Xi_{\mathcal{K}}$ contains the translation of each role inclusion occurring in $\mathcal{K}$, we have that $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$.

$R(s,t)$ is neither good nor aux-simple. Let $v_0, \ldots, v_n$, $\lambda_0, \ldots, \lambda_n$, and $\gamma_0, \ldots, \gamma_n$ be as determined in lines 10–12 when Algorithm 2 considers atom $R(s,t)$. For each $i \in [1..n]$, we have $\mathsf{pda}(\lambda_{i-1}, \gamma_{i-1}, \lambda_i, \gamma_i)$ is in $L(v_{i-1}, v_i)$ by line 13; thus there exists a role chain $\rho_i \in \mathcal{L}(\mathsf{pda}(\lambda_{i-1}, \gamma_{i-1}, \lambda_i, \gamma_i))$ such that $\rho_i(\nu(v_{i-1}), \nu(v_i)) \in \mathsf{inst}_I$. Next, we define ρ_0 by considering the following two cases, and in each case we will have that $\rho_0 \in \mathcal{L}(\mathsf{pda}(i_R, \bot, \lambda_0, \gamma_0))$ and $\rho_0(\nu(s), \nu(v_0)) \in \mathsf{inst}_I$.

- v_0 is a root of $\mathcal{S}$. Please note that, if $v \in \mathcal{V} \setminus \Psi_q$, then $v_0 \in \mathsf{r\text{-}ind}_{\mathsf{D}_{\mathcal{K}}}$ and $\tau(v_0) = \nu(v_0)$; otherwise, if $v_0 \in \mathcal{V} \cap \Psi_q$, then $\nu(v_0) \in \mathsf{r\text{-}ind}_{\mathsf{D}_{\mathcal{K}}}$. Because of the checks in lines 20 and 22 and by the definition of $\nu(v_0)$, we have that function $\mathsf{check}_{RI}(v_0, \nu(v_0), L)$ returns t, and so $\mathsf{D}_{\mathcal{K}} \cup \mathsf{rls}(i_R, \bot, \lambda_0, \gamma_0) \models S_{\mathcal{L}}(\tau(s), \nu(v_0))$ holds. Hence, by Lemma 9.10, there exist a role chain $\rho_0 = R_1 \cdots R_k$ with $\rho_0 \in \mathcal{L}(\mathsf{pda}(i_R, \bot, \lambda_0, \gamma_0))$ and constants $u_0, \ldots, u_k$ with $u_0 = \tau(s)$ and $u_k = \nu(v_0)$ such that $R_i(u_{i-1}, u_i) \in \mathsf{inst}_J$ for each $i \in [1..k]$; thus, $\rho_0(\tau(s), \nu(v_0)) \in \mathsf{inst}_J$. By properties (d3)–(d4) of Lemma 5.13, terms $w_0, \ldots, w_n$ with $w_0 = \nu(s)$ and $w_n = \nu(v_0)$ exist such that $R_i(w_{i-1}, w_i) \in \mathsf{inst}_I$ for each $i \in [1..n]$; and so $\rho_0(\nu(s), \nu(v_0)) \in \mathsf{inst}_I$.

- v_0 is not a root of $\mathcal{S}$. Hence $v_0 = s$ and $\tau(v_0)$ is an auxiliary constant. By line 17, a role chain ρ_0 exists with $\rho_0 \in \mathcal{L}(\mathsf{spda}(i_R, \bot, \lambda_0, \gamma_0, \tau(v_0)))$. By Definition 9.3 of stationary PDA, we have that $\rho_0 \in \mathcal{L}(\mathsf{pda}(i_R, \bot, \lambda_0, \gamma_0))$ and, for each role T occurring in ρ_0, we have $\mathbb{S}_T(\tau(v_0)) \in \mathsf{inst}_J$. By property (d2) of Lemma 5.13 and by the definition of ν, for each role T occurring in ρ_0, we have $T(\nu(v_0), \nu(v_0)) \in \mathsf{inst}_I$, and so $\rho_0(\nu(s), \nu(v_0)) \in \mathsf{inst}_I$.

Let $\rho' = \rho_0 \cdot \rho_1 \cdots \rho_n$. Clearly, we have that $\rho'(\nu(s), \nu(v_n)) \in \mathsf{inst}_I$ where $v_n = t$. Moreover, $\rho' \in \mathcal{L}(\mathsf{pda}(i_R, \bot, \lambda_n, \gamma_n))$ with $\lambda_n = f_R$ and $\gamma_n = \bot$. By Theorem, 8.3, we have $\rho' \in \mathcal{L}(R)$, and because $R(s,t)$ is not good, we have that $\top_r \notin \mathcal{L}(R)$. Since I is closed under $\Xi_{\mathcal{K}}$ and $\Xi_{\mathcal{K}}$ contains the translation of each role inclusion in $\mathcal{K}$, we have $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$. □

The soundness claim follows directly from Lemmas 9.11 and 6.17.

9.3.4 Completeness

Let π be a substitution such that $\mathsf{dom}(\pi) = \vec{x}$ and each element in $\mathsf{rng}(\pi)$ is an individual from $\mathsf{ind}_{\mathcal{K}}$, and assume that $\mathcal{K} \models_{\mathcal{DL}} \pi(q')$. By Proposition 5.2, we have that $\Xi_{\mathcal{K}} \models \pi(q')$; furthermore, by Theorem 2.3, a substitution π_* with $\mathsf{dom}(\pi_*) = \mathsf{var}(q')$ exists such that $\pi \subseteq \pi_*$ and $\|\pi_*(q')\|_I \subseteq \mathsf{inst}_I$. We next show that a candidate answer τ to q' over $\mathsf{D}_{\mathcal{K}}$ and a nondeterministic computation exist such that $\pi \subseteq \tau$ and $\mathsf{isSound}_{RI}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ returns t. In the following, we use essentially the same constructions as for the completeness proof in Section 6.4.

Let τ be substitution as defined in Definition 6.22 on page 107. Then, for each $x \in \vec{x}$, we have that $\tau(x) = \pi(x)$. By Lemmas 5.11 and 5.12, we have $\|\tau(q')\|_J \subseteq \mathsf{inst}_J$, and so $\mathsf{D}_{\mathcal{K}} \models \tau(q')$ due to Theorem 2.3. Hence, τ is a candidate answer to q' over $\mathsf{D}_{\mathcal{K}}$. Moreover, by Definition 6.1 of q, we also have that $\tau(q) \subseteq \mathsf{inst}_J$; furthermore, by Lemma 5.11 and the definition of τ, we have that $\pi_*(q) \subseteq \mathsf{inst}_I$. Finally note that, for each variable $y \in \mathsf{var}(q)$, we have that $\pi_*(y)$ is a labelled null, $\tau(y)$ is an auxiliary individual, and $\iota(\pi_*(y)) = \tau(y)$.

We are now ready to prove the completeness claim.

Lemma 9.12. *A nondeterministic computation exists such that the filtering procedure* $\mathsf{isSound}_{RI}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ *from Algorithm 2 returns* t.

Proof. By Lemma 6.24, function $\mathsf{isDSound}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ returns t. Hence, in the rest of this proof we consider the case in which an atom $R(s, t)$ exists in $q_\sim$ that is neither good nor aux-simple; otherwise, the algorithm returns t in line 2.

Let variable renaming σ be as specified in Definition 6.25. Then, for all distinct terms $s, t \in \mathsf{term}(\sigma(q_\sim))$, we have $\pi_*(s) \neq \pi_*(t)$. Also, due to $\pi_*(q_\sim) \subseteq \mathsf{inst}_I$, we have $\pi_*(\sigma(q_\sim)) \subseteq \mathsf{inst}_I$. Finally, we define the skeleton $\mathcal{S}$ for q, σ, and τ.

Furthermore, let substitution ν and skeleton $\mathcal{S} = \langle \mathcal{V}, \mathcal{E} \rangle$ be as specified in Definition 6.26 and let dir_I be the relation over the terms in inst_I specified just before Lemma 5.3 on page 55; moreover, let dir_I^+ be the transitive closure of dir_I. Recall that dir_I contains a pair $\langle w, w' \rangle$ for each edge $\langle w, w' \rangle$ in I that is direct for some role R and w' is a labelled null.

Since $\pi_* \subseteq \nu$, we clearly have that $\nu(\sigma(q_\sim)) \subseteq \mathsf{inst}_I$. Consider an arbitrary edge $\langle v, v' \rangle \in \mathcal{E}$. Due to $\langle \nu(v), \nu(v') \rangle \in \mathsf{dir}_I^+$, we have that terms $w_0, \ldots, w_k$, roles $R_1, \ldots, R_k$, and concepts $A_1, \ldots, A_k$ exist such that $w_0 = \nu(v)$, $w_k = \nu(v')$, and, for each $i \in [1..k]$, we have that $\mathbb{D}_R(w_{i-1}, w_i) \in \mathsf{inst}_I$ and w_i's type is R_i, A_i (i.e., $\iota(w_i) = o_{R_i,A_i}$). Note that all of these are uniquely defined by the edge. Then a role chain ρ is exist_{RI}*-compatible* with the edge $\langle v, v' \rangle$ if role chain $\chi_0, \ldots, \chi_k$ exist such that

- ρ is of the form $\rho = \chi_0 \cdot R_1 \cdot \chi_1 \cdots R_k \cdot \chi_k$, and
- for each $i \in [0..k]$ and each role T occurring in χ_i, we have that $\mathsf{Self}_T(\iota(w_i)) \in \mathsf{inst}_J$.

To prove the lemma, we show that the following two properties hold for each $v \in \mathcal{V}$.

($\diamondsuit$) For each $\langle v, v' \rangle \in \mathcal{E}$ and each PDA $\mathcal{P}_j \in L(v, v')$, a non-empty role chain $\rho_j \in \mathcal{L}(\mathcal{P}_j)$ exists that is exist_{RI}-compatible with the edge $\langle v, v' \rangle \in \mathcal{E}$.

($\heartsuit$) For each pair $\langle s, \mathcal{P}_j \rangle \in L(v)$ with s a term and $\mathcal{P}_j$ a PDA, a role chain $\rho \in \mathcal{L}(\mathcal{P}_j)$ exists such that $\rho(\tau(s), \nu(v)) \in \mathsf{inst}_J$.

Please note that, by lines 14 and 15 in Algorithm 2, we have that $\langle s, P \rangle \in L(v)$ implies that v is a root of $\mathcal{S}$. Hence, either $v \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ and $\tau(v) = \nu(v)$, or $v \in \Psi_q$ and $\nu(v) \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$. Therefore, by Lemma 9.9 and by the above notion of exist_{RI}-compatibility, and by Lemma 9.10, properties ($\heartsuit$) and ($\diamondsuit$) imply that

- for each $v \in \mathcal{V} \cap \Psi_q$, we have that $\mathsf{check}_{RI}(v, \nu(v), L)$ returns t in line 20, and
- for each $v \in \mathcal{V} \setminus \Psi_q$, we have that $\mathsf{check}_{RI}(v, \tau(v), L)$ returns t in line 22.

For the loop in line 6; let $R(s, t)$ be an aux-simple atom in $\sigma(q_\sim)$. By Definition 6.2 of aux-simple atom, we have $s \neq t$. By step 6, we have that $\mathsf{pda}(i_R, \bot, f_R, \bot) \in L(s, t)$. As $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$, a non-empty role chain $\rho = \chi_0 \cdot R_1 \cdot \chi_1 \cdots R_m \cdot \chi_m$ with $\rho \in \mathcal{L}(R)$ and terms $w_0, \ldots, w_m$ with $w_0 = \nu(s)$ and $w_m = \nu(t)$ exist that satisfy property (2) in Lemma 5.3. Since R is simple, we have that $|\rho| \leq 1$ and, by the definition of variable renaming σ, we also have that $\nu(s) \neq \nu(t)$; so $\chi_0 = \epsilon$. Thus $\rho = R_1$ is compatible with the edge $\langle s, t \rangle$.

For the loop in lines 7–17; let $R(s, t)$ be an atom in $\sigma(q_\sim)$ that is neither good nor aux-simple. We next determine the nondeterministic choices that preserve ($\diamondsuit$) in line

13, ($\heartsuit$) in line 15, and that do not satisfy the conditions in line 17. By assumption, we have $R(\nu(s), \nu(t)) \in \mathsf{inst}_I$. Thus, a non-empty role chain $\rho = \chi_0 \cdot R_1 \cdot \chi_1 \cdots R_m \cdot \chi_m$ with $\rho \in \mathcal{L}(R)$, and terms $w_0, \ldots, w_m$ with $w_0 = \nu(s)$ and $w_m = \nu(t)$ exist that satisfy property (2) in Lemma 5.3. To define vertex v_0 in line 9, we distinguish two cases, and for each we also define an index $\ell_0 \in [0..m]$ such that $w_{\ell_0} = \nu(v_0)$.

- If some $j \in [0..m]$ exists such that $w_j \in N_{\mathfrak{I}}$, let $v_0 = v_t$ and let ℓ_0 be the largest index such that $w_{\ell_0} = \nu(v_t)$.
- Otherwise, let $v_0 = s$ and let $\ell_0 = 0$.

Let $v_0, \ldots, v_n$ be the unique path connecting v_0 to t in $\mathcal{S}$. By the definition of ℓ_0 and the form of the terms $w_{\ell_0+1}, \ldots, w_m$, we have that $\nu(v_0) = w_{\ell_0}$ and $\nu(v_n) = w_m$; furthermore, for each $j \in [\ell_0 + 1..m]$, we have that $w_j \in \Phi_N$ and $\langle w_{j-1}, w_j \rangle \in \mathsf{dir}_I$. By Lemma 5.3, relation dir_I is a forest rooted in $N_{\mathfrak{I}}$; thus, for each $i \in [1..n]$, a unique index ℓ_i exists such that $\nu(v_i) = w_{\ell_i}$. Now let $\rho_0 = \chi_0 \cdots R_{\ell_0} \cdot \chi_{\ell_0}$, and let $\rho_i = R_{\ell_{i-1}+1} \cdot \chi_{\ell_{i-1}+1} \cdots R_{\ell_i} \cdot \chi_{\ell_i}$ for each $i \in [1..n]$; clearly, $\rho = \rho_0 \cdots \rho_n$. By properties (1) and (2) of Lemma 5.3 and by Lemma 5.12, role chain ρ_i is compatible with the edge $\langle v_{i-1}, v_i \rangle$ for each $i \in [1..n]$. Moreover, due to $\rho \in \mathcal{L}(R)$ and Theorem 8.3, we have that $\rho \in \mathcal{L}_{\mathsf{d}_\mathcal{R}}(\mathsf{pda}(i_R, \bot, f_R, \bot))$; therefore states $\lambda_0, \ldots, \lambda_n$ with $\lambda_n = f_R$ and words $\gamma_0, \ldots, \gamma_n$ with $\gamma_n = \bot$ exist such that $\rho_0 \in \mathcal{L}_{\mathsf{d}_\mathcal{R}}(\mathsf{pda}(i_R, \bot, \lambda_0, \gamma_0))$ and $\rho_i \in \mathcal{L}_{\mathsf{d}_\mathcal{R}}(\mathsf{pda}(\lambda_{i-1}, \gamma_{i-1}, \lambda_i, \gamma_i)$ for each $i \in [1..n]$. Since each ρ_i is compatible with $\langle v_{i-1}, v_i \rangle$, line 13 preserves property ($\diamondsuit$), as required. Finally, we consider the two cases in line 14.

- v_0 is a root of $\mathcal{S}$. Hence, if $v_0 \in \mathcal{V} \setminus \Psi_q$, then $v_0 \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ and $\tau(v_0) = \nu(v_0)$; otherwise, if $v_0 \in \mathcal{V} \cap \Psi_q$, then $\nu(v_0) \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$. By property (2) of Lemma 5.3, we have that $\rho_0(\nu(s), \nu(v_0)) \in \mathsf{inst}_I$. By Lemma 5.12, we have that $\rho_0(\tau(s), \nu(v_0)) \in \mathsf{inst}_J$, thus line 15 preserves property ($\heartsuit$).
- v_0 is not a root of $\mathcal{S}$. Hence $v_0 = s$ and s is a variable that τ maps to an auxiliary constant. By the definition of τ, we also have that $\iota(\nu(v_0)) = \tau(v_0)$. By the definition of ℓ_0, we have $\rho_0 = \chi_0$. By property (2) of Lemma 5.3, for each role T occurring in χ_0, we have $\mathbb{S}_T(\nu(v_0)) \in \mathsf{inst}_I$. Hence, by Lemma 5.12, for each role T

occurring in χ_0, we have $\mathbb{S}_T(\tau(v_0)) \in \mathsf{inst}_J$. By Definition 9.3 of stationary PDA and because $\rho_0 \in \mathcal{L}(\mathsf{pda}(i_R, \bot, \lambda_0, \gamma_0))$, we have $\rho_0 \in \mathcal{L}(\mathsf{spda}(i_R, \bot, \lambda_0, \gamma_0, \tau(v_0)))$, and so the condition in line 17 is not satisfied. □

Chapter 10

Acyclic Conjunctive Queries

Answering CQs over KBs formulated in the $\mathcal{EL}$ family of DLs is a computationally expensive task with the combined complexity of the problem ranging between NP-complete and PSPACE-complete depending on the form of role inclusions in the knowledge base. An exception are instance queries—CQs consisting of a single unary atom—which can be evaluated in polynomial time over all members of the $\mathcal{EL}$ family of DLs studied in this thesis. Unfortunately, the expressivity of instance queries is quite limited and many interesting queries do not fall within this class. For instance, the following Boolean query checks whether two objects exist that share a common ancestor, and cannot be expressed as an instance query.

$$q = \exists x, y, z.\, \mathsf{ancestor}(x, z) \wedge \mathsf{ancestor}(y, z) \tag{10.1}$$

The query in (10.1), however, belongs to a well-known class of Boolean CQs, the class of acyclic queries. Acyclic CQs generalise instance queries and can be answered in polynomial time over relational database. To the best of our knowledge, only two studies investigated the computational properties of acyclic CQs over KBs formulated in the $\mathcal{EL}$ family of DLs and they provide contrasting results: Bienvenu et al. [10] showed that acyclic CQ answering is tractable over $\mathcal{ELH}_\bot$ KBs, whereas Krötzsch et al. [66][1] showed that the problem is PSPACE-hard over $\mathcal{ELRO}^+_\bot$ KBs with regular RBoxes.

[1] The proof of the result is given in full details in the associated workshop paper [65]. While Krötzsch and Rudolph [65] do not explicitly consider acyclic CQs, a careful examination of the proof in Lemma 2 shows that the PSPACE lower complexity bound holds also for acyclic CQs.

In this chapter, we complete the complexity landscape of acyclic CQ answering for several important members of the $\mathcal{EL}$ family of DLs. To provide a more thorough investigation of the complexity of the problem, we start by introducing *arborescent queries*, a simple class of tree-shaped acyclic CQs in which all roles point towards the parent. Then, for $\mathcal{ELHO}_\bot$ knowledge bases, we show that answering arborescent queries can be done in polynomial time, while answering arbitrary acyclic CQs is intractable. We also show that answering arborescent queries is intractable over $\mathcal{EL}$ KBs that contain either a single transitive role or a single reflexive role. These lower bounds thus suggest that $\mathcal{ELH}_\bot$ is the largest member of the $\mathcal{EL}$ family for which acyclic CQ answering is tractable. Finally, we strengthen the known PSPACE lower bound of CQ answering over $\mathcal{ELRO}^+_\bot$ KBs by Krötzsch et al. [66] and show that the problem is PSPACE-hard in knowledge base complexity even when the CQ is an arborescent query and just the role inclusions are considered as part of the input (i.e., the query and all other parts of the knowledge base are fixed).

10.1 Acyclic and Arborescent Queries

Each Boolean CQ q can be associated with a directed graph dg_q, possibly containing loops, that describes the query's shape. In Definition 10.1, we define acyclic and arborescent queries by restricting the form of dg_q. Because our queries contain only unary and binary predicates, we follow Bienvenu et al. [10] and define acyclic CQs by checking that the undirected graph obtained from dg_q is acyclic. Nevertheless, our definition is consistent with the more general notion of acyclic CQs based on hypergraphs known from the theory of relational databases [38].

Definition 10.1. *For q a Boolean CQ, the directed graph $dg_q = \langle \mathsf{var}(q), E\rangle$ contains an edge $\langle x, y\rangle \in E$ for all variables x and y in $\mathsf{var}(q)$ for which a role R exists such that $R(x, y)$ is an atom in q. Then q is* acyclic *if the undirected graph obtained from dg_q by removing the orientation of edges is acyclic; furthermore, q is* arborescent *if dg_q is a directed, rooted tree in which all edges point towards the root.*

One can easily check that the query in (10.1) is arborescent, and so also acyclic. The next example illustrates the differences between acyclic and arborescent queries.

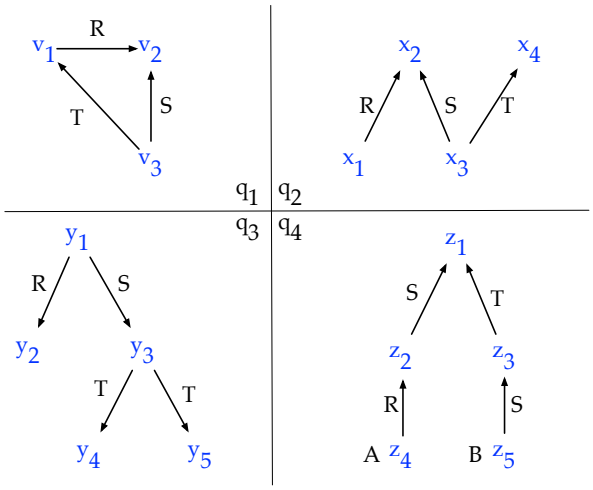

Figure 10.1: The directed graphs associated with the CQs from Example 10.1

Example 10.1. *Let q_1, q_2, q_3, and q_4 be the following Boolean CQs.*

$$q_1 = \exists \vec{v}.\, R(v_1, v_2) \land S(v_3, v_2) \land T(v_3, v_1) \tag{10.2}$$

$$q_2 = \exists \vec{x}.\, R(x_1, x_2) \land S(x_3, x_2) \land T(x_3, x_4) \tag{10.3}$$

$$q_3 = \exists \vec{y}.\, R(y_1, y_2) \land S(y_1, y_3) \land T(y_3, y_4) \land T(y_3, y_5) \tag{10.4}$$

$$q_4 = \exists \vec{z}.\, S(z_2, z_1) \land T(z_3, z_1) \land A(z_4) \land R(z_4, z_2) \land B(z_5) \land S(z_5, z_3) \tag{10.5}$$

Figure 10.1 shows the graph dg_{q_i} for each $i \in [1..4]$. Query q_1 is not acyclic as dg_{q_1} contains an undirected cycle, whereas all other queries are tree-shaped and thus acyclic. However, query q_2 is not rooted, whereas query q_3 is a rooted tree in which all edges point away from *(rather than towards) the root y_1; consequently, query q_4 is the only arborescent query.*

By using the 'rolling up' technique by Horrocks and Tessaris [52], tree-shaped queries of the form (10.4)—acyclic CQs in which the underlying directed graph is a rooted tree where all edges point *away* from the root—can be answered over a KB $\mathcal{K}$ in two steps. We first transform the CQ into a concept, and then check that the resulting concept is consistent with $\mathcal{K}$; for instance, CQ (10.4) can be equivalently expressed as the $\mathcal{EL}$ concept $\exists R.\top_c \sqcap \exists S.(\exists T.\top_c \sqcap \exists T.\top_c)$. Arborescent queries, however, can be used to test incoming connections from the root and the DLs in the $\mathcal{EL}$ family do not support inverse roles; therefore, the 'rolling up' technique is not applicable to arborescent queries over $\mathcal{EL}$ KBs.

10.2 Arborescent Queries over $\mathcal{ELHO}_\bot$ KBs

In Chapter 6 we presented a procedure that can be used to check whether a consistent $\mathcal{ELHO}_\bot$ knowledge base $\mathcal{K}$ entails an arborescent query q. We first compute the set of all candidate answers to q over $\mathsf{D}_\mathcal{K}$, and then we check whether at least one such candidate answer is sound. Even though by Theorem 6.13 we can check in polynomial time whether each candidate answer is sound w.r.t. $\mathcal{K}$, the number of candidate answers can be exponential in the size of q, so this procedure runs in exponential time in the worst-case. In this section, we improve this upper bound and present a tractable algorithm for answering arborescent queries over $\mathcal{ELHO}_\bot$ KBs. For convenience, our algorithm considers only a simplified form of arborescent queries that do not contain individuals. In Proposition 10.2, we exploit nominals and the fact that entailment of ground atoms over $\mathcal{ELHO}_\bot$ KBs can be decided in polynomial time to show that this restriction is without loss of generality.

Proposition 10.2. *For each $\mathcal{ELHO}_\bot$ knowledge base $\mathcal{K}$ and each arborescent query q over $\mathcal{K}$, one can compute in polynomial time an $\mathcal{ELHO}_\bot$ knowledge base $\mathcal{K}'$, an arborescent query q', and a conjunction of ground atoms q_{ind} such that*

- *q' does not contain individuals, and*
- *$\mathcal{K} \models_{\mathcal{DL}} q$ if and only if $\mathcal{K}' \models_{\mathcal{DL}} q'$ and $\mathcal{K}' \models_{\mathcal{DL}} q_{ind}$.*

Proof. Let $\mathcal{K}$ be an $\mathcal{ELHO}_\bot$ knowledge base and let q be an arborescent query over $\mathcal{K}$. We construct the arborescent query q', the conjunction of ground atoms q_{ind}, and the knowledge base $\mathcal{K}'$ in three steps.

We first construct the ground conjunctive query q_{ind}, and an arborescent query q_1 that does not contain ground atoms. To this end, let q_{ind} be the conjunctive query that contains all and only the ground atoms occurring in q, and let q_1 be the result of removing from q all its ground atoms. Clearly, q_1 is an arborescent query, and $\mathcal{K} \models_{\mathcal{DL}} q$ if and only if $\mathcal{K} \models_{\mathcal{DL}} q_1$ and $\mathcal{K} \models_{\mathcal{DL}} q_{ind}$.

At this point, if the arborescent query q_1 contains individuals, then these individuals must occur either in atoms of the form $R(x, a)$ or of the form $R(a, x)$. We next remove the former type of binary atoms from q_1. To this end, let q_2 be the result of replacing each

binary atom $R(x, a)$ in q_1 with the unary atom $A_{R,a}(x)$, where $A_{R,a}$ is a fresh atomic concept uniquely associated with $R(x, a)$. Because neither atoms of the form $R(x, a)$ nor unary atoms count towards determining whether a query is arborescent, query q_2 is arborescent as well. Next, let $\mathcal{K}_2$ be the knowledge base that extends $\mathcal{K}$ with an axiom $\exists R.\{a\} \sqsubseteq A_{R,a}$ for each binary atom $R(x, a)$ in q_1. By the definition of $\mathcal{K}_2$, each model $\mathcal{J}$ of $\mathcal{K}_2$ is also a model of $\mathcal{K}$; furthermore, for each fresh concept $A_{R,a}$, we have that $(\exists R.\{a\})^{\mathcal{J}} \subseteq (A_{R,a})^{\mathcal{J}}$. In addition, each model $\mathcal{I}$ of $\mathcal{K}$ can be expanded to a model $\mathcal{J}$ of $\mathcal{K}_2$ by interpreting each fresh concept $A_{R,a}$ as $(A_{R,a})^{\mathcal{J}} = (\exists R.\{a\})^{\mathcal{J}}$. By the definition of q_2, we then have that $\mathcal{K} \models_{\mathcal{DL}} q_1$ if and only if $\mathcal{K}_2 \models_{\mathcal{DL}} q_2$, as well as $\mathcal{K} \models_{\mathcal{DL}} q_{ind}$ if and only if $\mathcal{K}_2 \models_{\mathcal{DL}} q_{ind}$.

Finally, we remove from q_2 the binary atoms of the form $R(a, x)$. Towards this goal, let q' be the result of replacing each binary atom $R(a, x)$ with a binary atom $R(x_{R,a}, x)$ and a unary atom $A_a(x)$, where $x_{R,a}$ is a fresh variable uniquely associated with $R(a, x)$ and A_a is a fresh atomic concept uniquely associated with the individual a. Because each variable $x_{R,a}$ is fresh, the query q' is arborescent. Next, let $\mathcal{K}'$ be the knowledge base that extends $\mathcal{K}_2$ with a ground atom $A_a(a)$ for each fresh atomic concept A_a occurring in q'. By the definition of $\mathcal{K}'$, each model $\mathcal{J}$ of $\mathcal{K}'$ is also a model of $\mathcal{K}_2$; furthermore, for each fresh concept A_a, we have that $a^{\mathcal{J}} \in (A_a)^{\mathcal{J}}$. In addition, each model $\mathcal{I}$ of $\mathcal{K}_2$ can be expanded to a model $\mathcal{J}$ of $\mathcal{K}'$ by interpreting each fresh concept $A_{R,a}$ as $(A_{R,a})^{\mathcal{J}} = (\{a\})^{\mathcal{J}}$. By the definition of q', we then have that $\mathcal{K}_2 \models_{\mathcal{DL}} q_2$ if and only if $\mathcal{K}' \models_{\mathcal{DL}} q'$, as well as $\mathcal{K}_2 \models_{\mathcal{DL}} q_{ind}$ if and only if $\mathcal{K}' \models_{\mathcal{DL}} q_{ind}$. □

We next present our algorithm entails that, given a consistent $\mathcal{ELHO}_\bot$ KB $\mathcal{K}$ and an arborescent query q such that q does not contain individuals, decides whether $\mathcal{K} \models_{\mathcal{DL}} q$. Our algorithm uses the datalog program $\mathsf{D}_{\mathcal{K}}$ for $\mathcal{K}$ to check whether $\mathcal{K} \models_{\mathcal{DL}} q$. Arborescent queries, however, can contain 'forks'; therefore, $\mathsf{D}_{\mathcal{K}} \models q$ does not necessarily imply that $\mathcal{K} \models_{\mathcal{DL}} q$. We solve this problem by first computing in a top-down fashion all equality constraints for q—that is, all possible ways of identifying terms in the query due to 'forks'; then, we evaluate q bottom-up over $\mathsf{D}_{\mathcal{K}}$ using the derived constraints to avoid generating unsound answers.

Definition 10.3. *Let $\mathcal{K}$ be a consistent $\mathcal{ELHO}_\bot$ KB and let q be an arborescent query*

rooted in $r \in \mathsf{var}(q)$ *such that* q *does not contain individuals. Furthermore, let* $\mathsf{D}_\mathcal{K}$ *be the datalog program for* $\mathcal{K}$ *and let* $\mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ *and* $\mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ *be as specified in Definition 5.6.*

For each variable $y \in \mathsf{var}(q)$ *with* $y \neq r$*, and each set* $V \subseteq \mathsf{var}(q)$*, sets* rl_y *and* pred_V *are defined as follows.*

$$rl_y = \{R \in N_\mathfrak{R} \mid R(y,x) \in q \text{ with } x \text{ the parent of } y \text{ in the rooted tree } dg_q\}$$

$$\mathsf{pred}_V = \{y \in \mathsf{var}(q) \mid \exists x \in V \text{ with } x \text{ the parent of } y \text{ in the rooted tree } dg_q\}$$

Set RT *is the smallest set satisfying the following conditions.*

- $\{r\} \in \mathsf{RT}$ *and the level of* $\{r\}$ *is* 0*.*
- *For each set* $V \in \mathsf{RT}$ *with level* n *such that* $\mathsf{pred}_V \neq \emptyset$*, we have* $\mathsf{pred}_V \in \mathsf{RT}$ *and the level of* pred_V *is* $n+1$*.*
- *For each set* $V \in \mathsf{RT}$ *with level* n *and each* $y \in \mathsf{pred}_V$*, we have* $\{y\} \in \mathsf{RT}$ *and the level of* $\{y\}$ *is* $n+1$*.*

For each $V \in \mathsf{RT}$*, set* c_V *is defined as follows.*

$$\mathsf{c}_V = \{u \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}} \cup \mathsf{aux}_{\mathsf{D}_\mathcal{K}} \mid \mathsf{D}_\mathcal{K} \models B(u) \text{ for each atom } B(x) \in q \text{ with } x \in V\}$$

By reverse-induction on the level of the sets in RT*, each* $V \in \mathsf{RT}$ *is associated with a set* $\mathsf{A}_V \subseteq \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}} \cup \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$*.*

- *For each set* $V \in \mathsf{RT}$ *of maximal level, let* $\mathsf{A}_V = \mathsf{c}_V$*.*
- *For* $V \in \mathsf{RT}$ *a set of level* n *where* A_V *is undefined but* A_W *has been defined for each* $W \in \mathsf{RT}$ *of level* $n+1$*, let* $\mathsf{A}_V = \mathsf{c}_V \cap (\mathsf{i}_V \cup \mathsf{a}_V)$*, where* i_V *and* a_V *are as follows.*

$$\mathsf{i}_V = \{u \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}} \mid \forall y \in \mathsf{pred}_V \exists u' \in \mathsf{A}_{\{y\}} : \bigwedge_{R \in rl_y} \mathsf{D}_\mathcal{K} \models R(u',u)\}$$

$$\mathsf{a}_V = \{u \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}} \mid \exists u' \in \mathsf{A}_{\mathsf{pred}_V} \forall y \in \mathsf{pred}_V : \bigwedge_{R \in rl_y} \mathsf{D}_\mathcal{K} \models \mathbb{D}_R(u',u)\}$$

Function $\mathsf{entails}(\mathcal{K}, q)$ *returns* $\mathtt{t}$ *if and only if* $\mathsf{A}_{\{r\}}$ *is non-empty.*

The next theorem shows that function $\mathsf{entails}(\mathcal{K}, q)$ runs in polynomial time and correctly decides whether $\mathcal{K} \models_{\mathcal{DL}} q$.

Theorem 10.4. *For each consistent* $\mathcal{ELHO}_\bot$ *KB* $\mathcal{K}$ *and each arborescent query* q *such that* q *does not contain individuals,*

- *function* $\mathsf{entails}(\mathcal{K}, q)$ *returns* $\mathtt{t}$ *if and only if* $\mathcal{K} \models_{\mathcal{DL}} q$*, and*
- *function* $\mathsf{entails}(\mathcal{K}, q)$ *runs in time polynomial in the input size.*

Proof. Let $\mathcal{K}$ be a consistent $\mathcal{ELHO}_\bot$ KB and let q be an arborescent query such that q does not contain individuals. Furthermore, let $\Xi_\mathcal{K}$ and $\mathsf{D}_\mathcal{K}$ be the rule base and the datalog program for $\mathcal{K}$, respectively, and let I and J be universal interpretations for $\Xi_\mathcal{K}$ and $\mathsf{D}_\mathcal{K}$, respectively. Moreover, let ι be the function that maps the terms in I to constants as specified at the beginning of Section 5.3. Since q does not contain individuals, by Proposition 5.2 and Theorem 2.3 we have that $\mathcal{K} \models_{\mathcal{DL}} q$ if and only if a substitution π with $\mathsf{dom}(\pi) = \mathsf{var}(q)$ exists such that $\pi(q) \subseteq \mathsf{inst}_I$.

Before proving the theorem, we next define a couple of auxiliary notions. Let H be the height of the tree dg_q. Moreover, we uniquely associate each set $V \in \mathsf{RT}$ with the arborescent query q_V obtained from q by first removing each variable $z \in \mathsf{var}(q)$ and all atoms involving z such that $z \notin V$ and z is an ancestor of some $y \in V$ in the graph dg_q, and then replacing each variable $y \in V$ in the resulting query with a fresh variable y_V uniquely associated with V.

We are now ready to prove the theorem. We first argue that we can compute set RT in polynomial time. Let $\mathsf{RT}_0 = \{\{r\}\}$ and, for each $i \in [0..H-1]$, let

$$\mathsf{RT}_{i+1} = \{\mathsf{pred}_V \mid V \in \mathsf{RT}_i \wedge \mathsf{pred}_V \neq \emptyset\} \cup \{\{y\} \mid V \in \mathsf{RT}_i \wedge y \in \mathsf{pred}_V\}$$

By the definition of RT, we then have that $\mathsf{RT} = \bigcup_{i=0}^{H} \mathsf{RT}_i$; therefore, RT can be computed in polynomial time.

By Proposition 5.5 and Lemmas 5.12 and 5.13, each set $V \in \mathsf{RT}$ satisfies the two following properties for each constant $u \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}} \cup \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ and each term w such that $\iota(w) = u$.

(a) $u \in \mathsf{c}_V$ if and only if $B(w) \in I$ for each unary atom $B(y)$ occurring in q with $y \in V$.

(b) c_V can be computed in time polynomial in the size of $\mathcal{K}$ and q.

To prove the theorem, we next show that each set $V \in \mathsf{RT}$ satisfies the following two properties for each constant $u \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}} \cup \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$.

(1) $u \in \mathsf{A}_V$ if and only if a ground substitution π with $\mathsf{dom}(\pi) = \mathsf{var}(q_V)$ exists such that $\iota(\pi(y_V)) = u$ and $\pi(q_V) \subseteq \mathsf{inst}_I$.

(2) A_V can be computed in time polynomial in the size of $\mathcal{K}$ and q.

The proof is by reverse-induction on the level of the sets in RT.

Base case. Consider an arbitrary set $V \in \mathsf{RT}$ of level H. By the definition of A_V, we have $\mathsf{A}_V = \mathsf{c}_V$. Furthermore, an atom $B(y_V)$ is in q_V if and only if a variable $y \in V$ exists such that $B(y)$ is in q. Then properties (1) and (2) follow from properties (a) and (b).

Inductive step. Consider an arbitrary $n \in \mathbb{N}$. Let $V \in \mathsf{RT}$ be an arbitrary set of level n, and assume that properties (1) and (2) hold for each set $W \in \mathsf{RT}$ of level $n+1$; we show that V satisfies the properties.

Property (1). Let u be an arbitrary constant in $\mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}} \cup \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$. By the definition of A_V we have $\mathsf{A}_V = \mathsf{c}_V \cap (\mathsf{i}_V \cup \mathsf{a}_V)$. By the definition of q_V and from property (a), it suffices to show that $u \in (\mathsf{i}_V \cup \mathsf{a}_V)$ if and only if a substitution π exists such that $\pi(q_V) \subseteq \mathsf{inst}_I$ and $\iota(\pi(y_V)) = u$. We consider the two directions of (1) separately.

($\Rightarrow$) Assume that $u \in (\mathsf{i}_V \cup \mathsf{a}_V)$. We distinguish two cases.

- $u \in \mathsf{i}_V$. Thus, $u \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ and $u \in N_\mathfrak{I}$. Consider an arbitrary variable $y \in \mathsf{pred}_V$ and an arbitrary role $R \in rl_y$. Then, a constant $u' \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}} \cup \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ exists such that $u' \in \mathsf{A}_{\{y\}}$ and $R(u', u) \in \mathsf{inst}_J$. By the inductive hypothesis, a substitution $\pi_{\{y\}}$ exists such that $\pi_{\{y\}}(q_{\{y\}}) \subseteq \mathsf{inst}_I$ and $\iota(w') = u'$ where $w' = \pi_{\{y\}}(y_{\{y\}})$. By Lemma 5.13, we have $R(w', u) \in \mathsf{inst}_I$. Let π be the substitution such that $\pi(y_V) = u$ and $\pi_{\{y\}} \subseteq \pi$ for each $y \in \mathsf{pred}_V$. Then, we have $\pi(q_V) \subseteq \mathsf{inst}_I$.

- $u \in \mathsf{a}_V$. Thus, $u \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ and u is of the form $o_{P,A}$. It follows that a constant $u' \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}} \cup \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ exists such that $u' \in \mathsf{A}_{\mathsf{pred}_V}$ and $\mathbb{D}_R(u', u) \in \mathsf{inst}_J$ for each

$y \in \mathsf{pred}_V$ and each role $R \in rl_y$. By the inductive hypothesis, a substitution π_{pred_V} exists such that $\pi_{\mathsf{pred}_V}(q_{\mathsf{pred}_V}) \subseteq \mathsf{inst}_I$ and $\iota(w') = u'$, where $w' = \pi_{\mathsf{pred}_V}(y_{\mathsf{pred}_V})$. By Lemma 5.13, a term w^* with $\iota(w^*) = o_{P,A}$ exists in I such that $R(w', w^*) \in \mathsf{inst}_I$ for each $y \in \mathsf{pred}_V$ and each role $R \in rl_y$. Finally, let π be the substitution such that $\pi_{\mathsf{pred}_V} \subseteq \pi$, for each $y \in \mathsf{pred}_V$ we have $\pi(y) = w'$, and $\pi(y_V) = w^*$. Then, we have $\pi(q_V) \subseteq \mathsf{inst}_I$.

($\Leftarrow$) Assume that a substitution π exists such that $\pi(q_V) \subseteq \mathsf{inst}_I$ and $\iota(\pi(y_V)) = u$. We distinguish two cases.

- $\pi(y_V) \in N_\mathfrak{I}$. By Lemma 5.11, we have $u \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$ and $u \in N_\mathfrak{I}$. Consider an arbitrary variable $y \in \mathsf{pred}_V$ and an arbitrary role $R \in rl_y$. By the definition of q_V, atom $R(y, y_V)$ occurs in q_V. Since $R(\pi(y), \pi(y_V)) \in \mathsf{inst}_I$, by Lemma 5.12 we have that $R(\iota(\pi(y)), \iota(\pi(y_V))) \in \mathsf{inst}_J$ and $\iota(\pi(y)) \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}} \cup \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$. Due to the construction of q_V, $q_{\{y\}}$ is a subquery of q_V; thus, $\pi(q_{\{y\}}) \subseteq \mathsf{inst}_I$. Hence, by the inductive hypothesis, we have $\iota(\pi(y)) \in \mathsf{A}_{\{y\}}$. Due to $\iota(\pi(y_V)) = u$ and $u \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}}$, we have $u \in \mathsf{i}_V$.

- $\pi(y_V) \notin N_\mathfrak{I}$. By Lemma 5.11, we have $u \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ and u is of the form $o_{P,A}$. Consider an arbitrary variable $y \in \mathsf{pred}_V$ and an arbitrary role $R \in rl_y$. Then, atom $R(y, y_V)$ occurs in q_V and so $R(\pi(y), \pi(y_V)) \in \mathsf{inst}_I$; furthermore, by property (2) of Lemma 5.3 and because $\mathcal{K}$ is an $\mathcal{ELHO}_\bot$ knowledge base, we have that $P \sqsubseteq^*_\mathcal{R} R$ and $\mathbb{D}_P(\pi(y), \pi(y_V)) \in \mathsf{inst}_I$; thus $\langle \pi(y), \pi(y_V) \rangle \in \mathsf{dir}_I$. By property (1) of Lemma 5.3, relation dir_I is a forest rooted in $N_\mathfrak{I}$; so for each variable $z \in \mathsf{pred}_V$, we have that $\pi(y) = \pi(z)$. Furthermore, by Lemma 5.12, we have that $\mathbb{D}_P(\iota(\pi(y)), \iota(\pi(y_V))) \in \mathsf{inst}_J$ and $\iota(\pi(y)) \in \mathsf{r\text{-}ind}_{\mathsf{D}_\mathcal{K}} \cup \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$. Since $P \sqsubseteq^*_\mathcal{R} R$ and J is closed under $\mathsf{D}_\mathcal{K}$, we have that $\mathbb{D}_R(\iota(\pi(y)), \iota(\pi(y_V))) \in \mathsf{inst}_J$. Let π_{pred_V} be the substitution that extends π by setting $\pi_{\mathsf{pred}_V}(y_{\mathsf{pred}_V}) = \pi(y)$. Then, we have that $\pi_{\mathsf{pred}_V}(q_{\mathsf{pred}_V}) \subseteq \mathsf{inst}_I$ and, by the inductive hypothesis, we have that $\iota(\pi(y)) \in \mathsf{A}_{\mathsf{pred}_V}$. Due to $\iota(\pi(y_V)) = u$ and $u \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$, we have $u \in \mathsf{a}_V$.

Property (2). We show that A_V can be computed in time polynomial in the size of $\mathsf{D}_\mathcal{K}$ and q. By property (b), set c_V can be computed in time polynomial in q and $\mathsf{D}_\mathcal{K}$. But then,

by Proposition 5.5 sets i_V and a_V can also be computed in polynomial time, and so A_V can be computed in polynomial time as well. □

By Proposition 5.7, we can use program $\mathsf{D}_\mathcal{K}$ to check the consistency of an $\mathcal{ELHO}_\bot$ KB $\mathcal{K}$ in polynomial time, whereas by Proposition 10.2 we can use our algorithm entails to answer arbitrary arborescent queries; thus the following result follows directly from Theorem 10.4.

Theorem 10.5. *For an $\mathcal{ELHO}_\bot$ knowledge base $\mathcal{K}$ and an arborescent query q, checking $\mathcal{K} \models_{\mathcal{DL}} q$ is* PTIME*-complete in combined complexity.*

10.3 Lower Bounds for Acyclic Queries

We next show that, unless PTIME equals NP, answering arbitrary acyclic queries over $\mathcal{ELHO}_\bot$ KBs is harder than answering arborescent queries; moreover, we show that adding transitive or reflexive roles to the DL $\mathcal{EL}$ makes answering arborescent queries intractable.

Theorem 10.6. *For a knowledge base $\mathcal{K} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$ and a Boolean CQ q, checking $\mathcal{K} \models_{\mathcal{DL}} q$ is* NP*-hard in combined complexity in each of the following cases.*

1. *The CQ q is acyclic and the KB $\mathcal{K}$ is in $\mathcal{ELHO}_\bot$.*
2. *The CQ q is arborescent, the ABox $\mathcal{A}$ contains a single unary assertion, the TBox $\mathcal{T}$ contains only axioms of the form $A_1 \sqsubseteq A$ and $A_1 \sqsubseteq \exists R.A$, and the RBox $\mathcal{R}$ contains a single axiom of the form $R \cdot R \sqsubseteq R$.*
3. *The CQ q is arborescent, the ABox $\mathcal{A}$ contains a single unary assertion, the TBox $\mathcal{T}$ contains only axioms of the form $A_1 \sqsubseteq A$ and $A_1 \sqsubseteq \exists R.A$, and the RBox $\mathcal{R}$ contains a single axiom of the form $\epsilon \sqsubseteq R$.*

Proof. All lower bounds are proved by reducing the NP-hard problem of checking the satisfiability of a CNF formula φ [33]. For the rest of this proof, we fix a CNF formula $\varphi = \bigwedge_{j=1}^{m} C_j$ where each C_j is a clause over variables $\{v_1, \dots, v_n\}$. In the following, given two Boolean CQs q and q' with $\mathsf{var}(q) \cap \mathsf{var}(q') = \emptyset$, let $q \wedge q'$ be the Boolean CQ that contains all atoms occurring in q and all atoms occurring in q'.

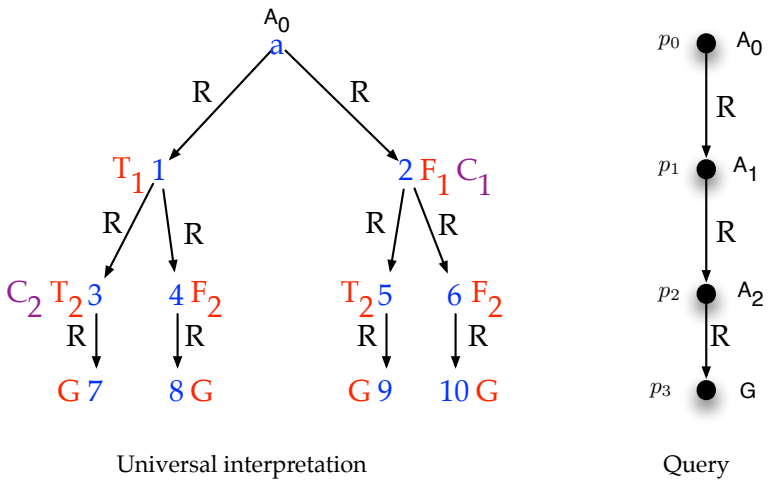

Figure 10.2: Universal interpretation I_0 and query q_0

Then to prove the theorem, we will first define a KB $\mathcal{K}_0$ containing only axioms of the form $A_1 \sqsubseteq A$ and $A_1 \sqsubseteq \exists R.A$, and a Boolean CQ q_0, after which we will construct

(1) a KB $\mathcal{K}_1$ and an acyclic CQ q_1 such that $\mathcal{K}_1$ is in $\mathcal{ELHO}_\bot$, and φ is satisfiable if and only if $\mathcal{K}_0 \cup \mathcal{K}_1 \models_{\mathcal{DL}} q_0 \wedge q_1$,

(2) a KB $\mathcal{K}_2$ and an arborescent CQ q_2 such that $\mathcal{K}_2$ contains a single axiom of the form $R \cdot R \sqsubseteq R$ and φ is satisfiable if and only if $\mathcal{K}_0 \cup \mathcal{K}_2 \models_{\mathcal{DL}} q_0 \wedge q_2$, and

(3) a KB $\mathcal{K}_3$ and an arborescent CQ q_3 such that $\mathcal{K}_3$ contains a single axiom of the form $\epsilon \sqsubseteq R$ and φ is satisfiable if and only if $\mathcal{K}_0 \cup \mathcal{K}_3 \models_{\mathcal{DL}} q_0 \wedge q_3$.

Construction of $\mathcal{K}_0$ and q_0. In our construction, we use a fresh role R, a fresh concept G, fresh concepts T_i, F_i, and A_i uniquely associated with each variable v_i in φ, and fresh concepts C_j uniquely associated to each clause C_j in φ.

Then, KB $\mathcal{K}_0$ contains an assertion (10.6), and axioms (10.7)–(10.13). We next describe how the axioms in $\mathcal{K}_0$ model the structure of a universal interpretation I_0 of $\mathcal{K}_0$. Assertion (10.6) and axioms (10.7)–(10.11) encode a binary tree of depth $n+1$ in I_0 rooted in individual a in which edges are labelled by role R, each leaf node satisfies concept G, and each node w at depth $1 \leq i \leq n$ satisfies exactly one of T_i and F_i. Then node w represents a positive truth assignment to v_i if $T_i(w) \in I_0$; otherwise, it represents a negative truth assignment to v_i. Consequently, a path from a to a leaf node in the tree represents a truth assignment

to the variables in φ. Finally, axioms (10.12) and (10.13) ensure that, for each node w at depth $1 \leq i \leq n$ in the tree, if the truth assignment to v_i represented by w makes clause C_j evaluate to true, then $C_j(w) \in I_0$. The left part of Figure 10.2 shows a universal interpretation of $\mathcal{K}_0$ for a CNF formula with 2 variables.

$$A_0(a) \tag{10.6}$$

$$A_{i-1} \sqsubseteq \exists R.T_i \qquad \forall i \in [1..n] \tag{10.7}$$

$$A_{i-1} \sqsubseteq \exists R.F_i \qquad \forall i \in [1..n] \tag{10.8}$$

$$T_i \sqsubseteq A_i \qquad \forall i \in [1..n] \tag{10.9}$$

$$F_i \sqsubseteq A_i \qquad \forall i \in [1..n] \tag{10.10}$$

$$A_n \sqsubseteq \exists R.G \tag{10.11}$$

$$T_i \sqsubseteq C_j \qquad \forall i \in [1..n]\ \forall j \in [1..m] \text{ with } v_i \in C_j \tag{10.12}$$

$$F_i \sqsubseteq C_j \qquad \forall i \in [1..n]\ \forall j \in [1..m] \text{ with } \neg v_i \in C_j \tag{10.13}$$

Query q_0 is given in (10.14) where $\vec{p} = p_0, \ldots, p_{n+1}$ are fresh variables.

$$q_0 = \exists \vec{p}.\ \bigwedge_{i=0}^{n} [A_i(p_i) \wedge R(p_i, p_{i+1})] \wedge G(p_{n+1}) \tag{10.14}$$

The right part of Figure 10.2 shows the query q_0 for a CNF formula with 2 variables. It should be clear that, for each substitution π with $\mathsf{dom}(\pi) = \vec{p}$ such that $\pi(q_0) \subseteq I_0$, we have that $\pi(p_0) = a$ and $\pi(p_{n+1})$ is a leaf; furthermore, there exists a one-to-one correspondence between the truth assignments for φ and the ground substitutions embedding q_0 into I_0.

Construction of $\mathcal{K}_1$ and q_1. In our construction, we use fresh roles S_j and individuals c_j uniquely associated with each clause C_j of φ.

Then KB $\mathcal{K}_1$ contains assertions (10.15), and axioms (10.16) and (10.17). Let I_1 be a universal interpretation of $\mathcal{K}_0 \cup \mathcal{K}_1$. We next describe how the axioms and assertions in $\mathcal{K}_1$ modify the tree encoded by $\mathcal{K}_0$. Axioms (10.16) ensure that, for each node w in the tree whose truth assignment makes clause C_j evaluate to true, there exists an edge labelled by S_j from w to individual c_j. Assertion (10.15) generates an edge labelled by R connecting c_j

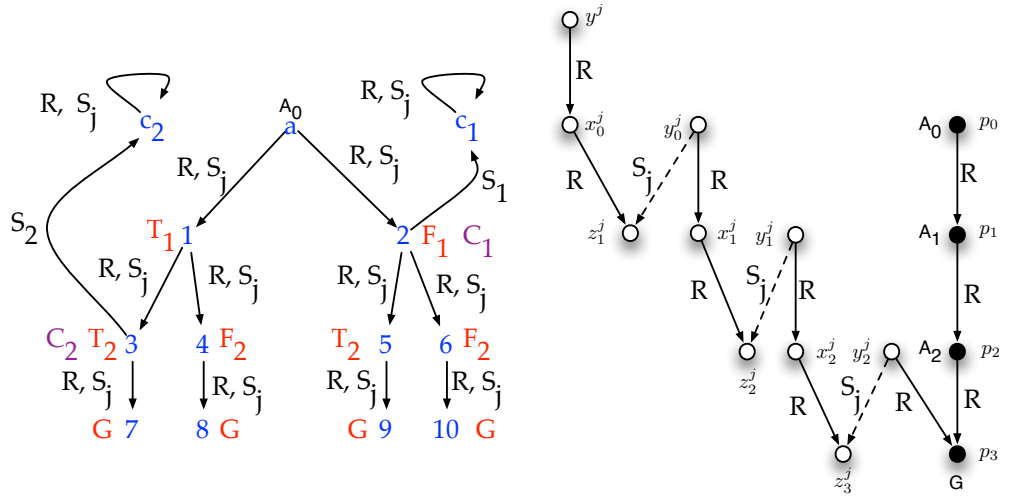

Figure 10.3: Universal interpretation I_1 and query $q_0 \wedge q_1$

to itself. Finally, axiom (10.17) labels each edge in the tree and each looping edge with S_j. Figure 10.3 shows the universal interpretation I_1 obtained from the universal interpretation I_0 in Figure 10.2 by applying the axioms in $\mathcal{K}_1$.

$$R(c_j, c_j) \qquad \forall j \in [1..m] \tag{10.15}$$

$$C_j \sqsubseteq \exists S_j.\{c_j\} \qquad \forall j \in [1..m] \tag{10.16}$$

$$R \sqsubseteq S_j \qquad \forall j \in [1..m] \tag{10.17}$$

In our construction of q_1, we use use variable p_{n+1} from (10.14), as well as fresh variables $\vec{x} = x_0^j, \ldots, x_n^j$, $\vec{y} = y^j, y_0^j, \ldots, y_n^j$, and $\vec{z} = z_1^j, \ldots, z_{n+1}^j$ uniquely associated with each clause C_j. Then, let $q_1 = \exists\vec{x}\,\exists\vec{y}\,\exists\vec{z}.\ \psi^1 \wedge \cdots \wedge \psi^m$ where, for each clause C_j, the conjunction of atoms ψ^j is as specified in (10.18)–(10.21).

$$\psi^j = R(y^j, x_0^j) \tag{10.18}$$

$$\wedge \bigwedge_{i=0}^{n} R(x_i^j, z_{i+1}^j) \wedge S_j(y_i^j, z_{i+1}^j) \tag{10.19}$$

$$\wedge \bigwedge_{i=0}^{n-1} R(y_i^j, x_{i+1}^j) \tag{10.20}$$

$$\wedge R(y_n^j, p_{n+1}) \tag{10.21}$$

Figure 10.3 shows query $q_0 \wedge q_1$ for $n = 2$ and an arbitrary clause C_j. It should be clear that query $q_0 \wedge q_1$ is acyclic, as required; in contrast $q_0 \wedge q_1$ is not arborescent: each variable y_i^j in $q_0 \wedge q_1$ has two parent nodes. We next prove that φ is satisfiable if and only if $\mathcal{K}_0 \cup \mathcal{K}_1 \models_{\mathcal{DL}} q_0 \wedge q_1$.

($\Rightarrow$) Consider an arbitrary truth assignment $\nu : \{v_1, \ldots, v_n\} \to \{\mathsf{t}, \mathsf{f}\}$ and assume that $\nu(\varphi) = \mathsf{t}$; we show that a substitution π with $\mathsf{dom}(\pi) = \mathsf{var}(q_0 \wedge q_1)$ exists such that $\pi(q_0 \wedge q_1) \subseteq I_1$. To this end, let $\pi(p_0) = a$ and, for each $i \in [1..n]$, let $\pi(p_i)$ be the unique successor of $\pi(p_{i-1})$ in I_1 such that $T_i(\pi(p_i)) \in I_1$ if and only if $\nu(v_i) = \mathsf{t}$. Consider an arbitrary clause C_j. Since $\nu(\varphi) = \mathsf{t}$, a literal $l \in C_j$ exists such that $\nu(l) = \mathsf{t}$. Let $k \in [1..n]$ be the unique index such that literal l is over variable v_k. Then, we extend substitution π as follows:

- let $\pi(y^j) = c_j$ and for each $\ell \in [0..k-1]$, let $\pi(x_\ell^j) = \pi(y_\ell^j) = \pi(z_{\ell+1}^j) = c_j$, and
- let $\pi(x_k^j) = c_j$, let $\pi(z_{k+1}^j) = c_j$, and let $\pi(y_k^j) = \pi(p_k)$, and
- for each $\ell \in [k+1..n]$, let $\pi(x_\ell^j) = \pi(y_\ell^j) = \pi(p_\ell)$, and let $\pi(z_{\ell+1}^j) = \pi(p_{\ell+1})$.

Due to the atoms (10.15) and the axioms (10.16)–(10.17), one can check that $\pi(\psi^j) \subseteq I_1$; thus $\pi(q_0 \wedge q_1) \subseteq I_1$, as required.

($\Leftarrow$) Consider an arbitrary ground substitution π and assume that $\pi(q_0 \wedge q_1) \subseteq I_1$; we next show that a truth assignment $\nu : \{v_1, \ldots, v_n\} \to \{\mathsf{t}, \mathsf{f}\}$ exists such that $\nu(\varphi) = \mathsf{t}$. By the definition of $\mathcal{K}_0$ and q_0, we have that $\pi(p_0) = a$; furthermore, for each $i \in [1..n]$, we have that $\pi(p_i)$ is a successor of $\pi(p_{i-1})$ in the binary tree encoded by $\mathcal{K}_0$. Then, let ν be the truth assignment such that, for each $i \in [1..n]$, we have $\nu(v_i) = \mathsf{t}$ if and only if $T_i(\pi(p_i)) \in I_1$. We next show that $\nu(\varphi) = \mathsf{t}$; that is, for each clause C_j, we have $\nu(C_j) = \mathsf{t}$. Consider an arbitrary clause C_j. By the definition of $\mathcal{K}_0$, it suffices to find an index $\ell \in [1..n]$ such that $C_j(\pi(p_\ell)) \in I_1$. To this end, we show that substitution π satisfies the following property for each $i \in [0..n]$.

($\diamondsuit$) If for each $\ell \in [i..n]$ we have $\pi(z_{\ell+1}^j) \notin N_{\mathfrak{I}}$, then $\pi(y_i^j) = \pi(p_i)$ and $\pi(x_i^j) = \pi(p_i)$.

We proceed by reverse induction on $i \in [0..n]$.

Base case. Let $i = n$. Assume that $\pi(z^j_{n+1}) \notin N_{\mathfrak{I}}$. By the definition of q_1, we have that $\pi(y^j_n)$ is the unique parent of $\pi(p_{n+1})$ in the tree encoded by $\mathcal{K}_0$, and so $\pi(y^j_n) = \pi(p_n)$. Due to atom $S_j(y^j_n, z^j_{n+1})$ in ψ^j, we have $\pi(z^j_{n+1})$ is a successor of $\pi(p_n)$. Due to atom $R(x^j_n, z^j_{n+1})$ in ψ^j, we have $\pi(x^j_n)$ is the parent of $\pi(z^j_{n+1})$, thus $\pi(x^j_n) = \pi(p_n)$, as required.

Inductive step. Consider an arbitrary $i \in [0..n-1]$, and assume that the property holds for $i+1$, and that $\pi(z^j_{\ell+1}) \notin N_{\mathfrak{I}}$ for each $\ell \in [i..n]$. By the inductive hypothesis, we have $\pi(y^j_{i+1}) = \pi(p_{i+1})$ and $\pi(x^j_{i+1}) = \pi(p_{i+1})$. Due to atom $R(y^j_i, x^j_{i+1})$ in ψ^j, we have $\pi(y^j_i)$ is the unique parent of $\pi(p_{i+1})$ in the tree, and so $\pi(y^j_i) = \pi(p_i)$. Due to atom $S_j(y^j_i, z^j_{i+1})$ in ψ^j, we have that $\pi(z^j_{i+1})$ is a successor of $\pi(p_i)$. Finally, due to atom $R(x^j_i, z^j_{i+1})$ in ψ^j, we have $\pi(x^j_i) = \pi(p_i)$.

We next show that there exists $i \in [1..n]$ such that $\pi(z^j_{i+1})$ is mapped to an individual from $N_{\mathfrak{I}}$. Assume the opposite, hence, for each $i \in [1..n]$, we have that $\pi(z^j_{i+1}) \notin N_{\mathfrak{I}}$. By property $(\diamondsuit)$, we then have that $\pi(x^j_1) = \pi(p_1)$ and, because atom $R(y^j_0, x^j_1)$ occurs in q_1, we also have that $\pi(y^j_0) = \pi(p_0) = a$. We next distinguish two cases.

- $\pi(z^j_1) \in N_{\mathfrak{I}}$. Because $\pi(y^j_0) = a$, atom $S_j(y^j_0, z^j_1)$ occurs in q_1, and a is not connected to an individual in I_1, we have $S_j(\pi(y^j_0), \pi(z^j_1)) \notin I_1$, which is a contradiction.
- $\pi(z^j_1) \notin N_{\mathfrak{I}}$. Then, by property $(\diamondsuit)$, we also have that $\pi(x^j_0) = \pi(p_0) = a$. Since q_1 contains an atom $R(y^j, x^j_0)$ and individual a does not have incoming connections in I_1, we have that $R(\pi(y^j), \pi(x^j_0)) \notin I_1$, which is a contradiction.

Let $\ell \in [1..n]$ be the largest index such that $\pi(z^j_{\ell+1}) \in N_{\mathfrak{I}}$. By property $(\diamondsuit)$, we then have that $\pi(y^j_\ell) = \pi(p_\ell)$. By axioms (10.16) and (10.17) in $\mathcal{K}_1$, and because atom $S_j(y^j_\ell, z^j_{\ell+1})$ occurs in q_1, we have $\pi(z^j_{\ell+1}) = c_j$ and $C_j(\pi(p_\ell)) \in I_1$, as required.

Construction of $\mathcal{K}_2$ and q_2. The KB $\mathcal{K}_2$ consists only of a single transitivity axiom (10.22). Let I_2 be a universal interpretation of $\mathcal{K}_0 \cup \mathcal{K}_2$. Then axiom (10.22) modifies the tree encoded by $\mathcal{K}_0$ by connecting each node w in the tree via an R edge to all nodes that occur on the path connecting w to the root a. Figure 10.4 shows the universal interpretation I_2 obtained from the universal interpretation I_0 in Figure 10.2 by applying the role inclusion

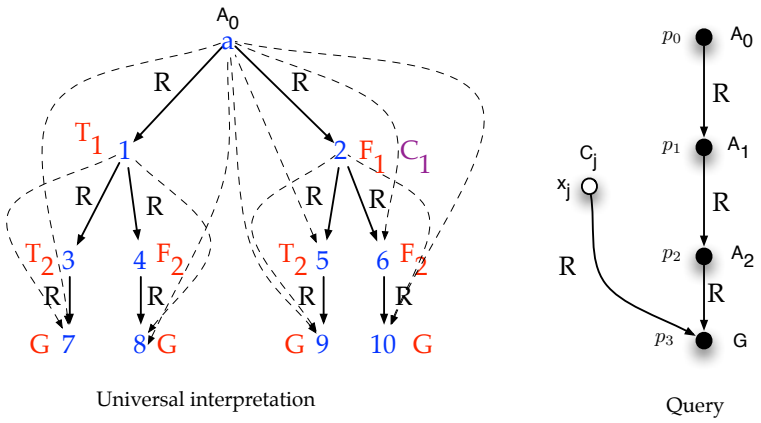

Figure 10.4: Universal interpretation I_2 and query $q_0 \wedge q_2$

in $\mathcal{K}_2$; the edges obtained using the transitivity of role R are dashed.

$$R \cdot R \sqsubseteq R \tag{10.22}$$

In our construction of q_2, we use variable p_{n+1} from query q_0, as well as a fresh variable x_j uniquely associated to each clause C_j. Query q_2 is given in (10.23).

$$q_2 = \exists x_1 \ldots \exists x_m. \bigwedge_{i=1}^{m} C_j(x_j) \wedge R(x_j, p_{n+1}) \tag{10.23}$$

Figure 10.4 shows query $q_0 \wedge q_2$ for $n = 2$ and an arbitrary clause C_j. It should be clear that $q_0 \wedge q_2$ is arborescent. We next prove that φ is satisfiable if and only if $\mathcal{K}_0 \cup \mathcal{K}_2 \models_{\mathcal{DL}} q_0 \wedge q_2$.

($\Rightarrow$) Consider an arbitrary truth assignment $\nu : \{v_1, \ldots, v_n\} \to \{\mathsf{t}, \mathsf{f}\}$ and assume that $\nu(\varphi) = \mathsf{t}$; we show that a substitution π with $\mathsf{dom}(\pi) = \mathsf{var}(q_0 \wedge q_2)$ exists such that $\pi(q_0 \wedge q_2) \subseteq I_2$. To this end, let $\pi(p_0) = a$ and, for each $i \in [1..n]$, let $\pi(p_i)$ be the unique successor of $\pi(p_{i-1})$ in I_2 such that $T_i(\pi(p_i)) \in I_2$ if and only if $\nu(v_i) = \mathsf{t}$. Since $\nu(\varphi) = \mathsf{t}$, for each clause C_j, a literal $l \in C_j$ exists such that $\nu(l) = \mathsf{t}$; then, we let $\pi(x_j) = \pi(p_k)$ where $k \in [1..n]$ is the unique index such that literal l is over variable v_k. Due to axioms (10.12) and (10.13) in $\mathcal{K}_0$ and because role R is transitive, for each $j \in [1..m]$, we then have that $\{C_j(\pi(x_j)), R(\pi(x_j), \pi(p_{n+1})\} \subseteq I_2$; thus $\pi(q_0 \wedge q_2) \subseteq I_2$.

($\Leftarrow$) Consider an arbitrary ground substitution π and assume that $\pi(q_0 \wedge q_2) \subseteq I_2$; we next show that a truth assignment $\nu : \{v_1, \ldots, v_n\} \to \{\mathsf{t}, \mathsf{f}\}$ exists such that $\nu(\varphi) = \mathsf{t}$. By

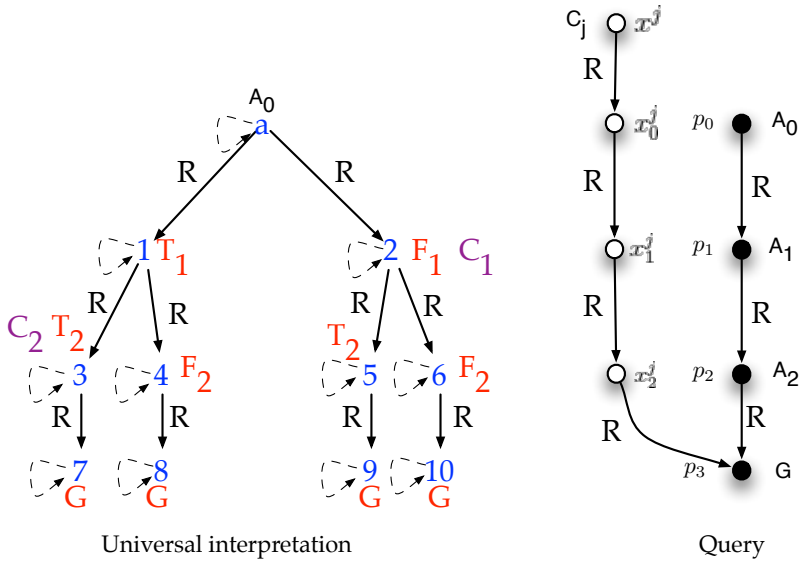

Figure 10.5: Universal interpretation I_3 and query $q_0 \wedge q_3$

the definition of q_0 and q_2, for each clause C_j occurring in φ, term $\pi(x_j)$ occurs on the unique path connecting $\pi(p_{n+1})$ to individual a and $C_j(\pi(x_j)) \in I_2$; therefore, an index $\ell_j \in [1..n]$ exists such that $\pi(p_{\ell_j}) = \pi(x_j)$. By the definition of $\mathcal{K}_0$ and q_0, the truth assignment represented by π then makes each clause C_j of φ evaluate to true.

Construction of $\mathcal{K}_3$ and q_3. The KB $\mathcal{K}_3$ consists of a single reflexivity axiom (10.24). Let I_3 be a universal interpretation of $\mathcal{K}_0 \cup \mathcal{K}_3$. Axiom (10.24) modifies the tree encoded by $\mathcal{K}_0$ by generating an edge labelled by R connecting each node in the tree to itself. Figure 10.5 shows the universal interpretation I_3 obtained from I_0 in Figure 10.2 by applying the role inclusion in $\mathcal{K}_3$; the edges obtained using the reflexivity of role R are dashed.

$$\epsilon \sqsubseteq R \tag{10.24}$$

In our construction of q_3, we use variable p_{n+1} from query q_0, as well as fresh variables $\vec{x} = x_0^j, \ldots, x_n^j$ uniquely associated to each clause C_j. Then, let $q_3 = \exists \vec{x}.\, \psi^1 \wedge \ldots \wedge \psi^m$ where, for each clause C_j, the formula ψ^j is as specified in (10.25).

$$\psi^j = C_j(x_0^j) \wedge R(x^j, x_0^j) \wedge \bigwedge_{i=1}^{n} R(x_{i-1}^j, x_i^j) \wedge R(x_n^j, p_{n+1}) \tag{10.25}$$

Figure 10.5 shows query $q_0 \wedge q_3$ for $n = 2$ and an arbitrary clause C_j. It should be clear that $q_0 \wedge q_3$ is arborescent. We next show that φ is satisfiable if and only if $\mathcal{K}_0 \cup \mathcal{K}_3 \models_{\mathcal{DL}} q_0 \wedge q_3$.

($\Rightarrow$) Consider an arbitrary truth assignment $\nu : \{v_1, \ldots, v_n\} \to \{\mathsf{t}, \mathsf{f}\}$ and assume that $\nu(\varphi) = \mathsf{t}$; we next show that a substitution π with $\mathsf{dom}(\pi) = \mathsf{var}(q_0 \wedge q_3)$ exists such that $\pi(q_0 \wedge q_3) \subseteq I_3$. To this end, let $\pi(p_0) = a$ and, for each $i \in [1..n]$, let $\pi(p_i)$ be the unique successor of $\pi(p_{i-1})$ in I_3 such that $T_i(\pi(p_i)) \in I_3$ if and only if $\nu(v_i) = \mathsf{t}$. Please observe that $\pi(q_0) \subseteq I_3$. Consider an arbitrary clause C_j of φ. Since $\nu(\varphi) = \mathsf{t}$, a literal $l \in C_j$ and a variable v_k exist such that $\nu(l) = \mathsf{t}$ and l is over v_k. Then, for each $i \in [0..k]$, we let $\pi(x_i^j) = \pi(p_k)$; furthermore, for each $\ell \in [k+1..n]$, we let $\pi(x_\ell^j) = \pi(p_\ell)$. Due to axioms (10.12) and (10.13) in $\mathcal{K}_0$, we have that $C_j(\pi(x_0^j) \in I_3$; furthermore, because role R is reflexive and $\pi(q_0) \subseteq I_3$, for each $i \in [1..n]$, we have that $R(\pi(x_{i-1}^j), \pi(x_i^j)) \in I_3$. Thus, for each clause C_j, we have that $\pi(\psi^j) \subseteq I_3$, and so $\pi(q_0 \wedge q_3) \subseteq I_3$.

($\Leftarrow$) Consider an arbitrary ground substitution π and assume that $\pi(q_0 \wedge q_2) \subseteq I_2$; we next show that a truth assignment $\nu : \{v_1, \ldots, v_n\} \to \{\mathsf{t}, \mathsf{f}\}$ exists such that $\nu(\varphi) = \mathsf{t}$. For each clause C_j, formula ψ^j ensures that a path of length $n+1$ exists connecting $\pi(p_{n+1})$ to a node $\pi(x_0^j)$ occurring in the unique path connecting $\pi(p_{n+1})$ to a in I_3. Since the tree has depth $n+1$ and the leaves and the root of the tree do no satisfy concept C_j, variable $\pi(x_0^j)$ must be at depth $1 \leq i \leq n$. By the definition of $\mathcal{K}_0$ and q_0, the truth assignment represented by π then makes each clause C_j evaluate to true, and thus φ is satisfiable. $\square$

In Chapter 9, we presented a CQ answering algorithm for $\mathcal{ELRO}_\bot^+$ that uses space polynomial in the total size of the input. This algorithm is worst-case optimal in combined complexity: Krötzsch et al. [66] proved this by reducing the PSPACE-hard problem of checking nonemptiness of the intersection of the languages generated by m deterministic finite automata $\mathcal{F}_1, \ldots, \mathcal{F}_m$ over a common alphabet Σ [62] to acyclic CQ answering in $\mathcal{ELRO}_\bot^+$. In the knowledge base $\mathcal{K}$ encoding the problem, a regular RBox contains roles $R_1 \ldots R_m$ such that $\mathcal{L}(R_i) = \mathcal{L}(\mathcal{F}_i)$ for each $i \in [1..m]$; furthermore, a TBox ensures that a universal interpretation of $\mathcal{K}$ is a rooted tree so, for each $\rho \in \Sigma^*$, a term w_ρ exists that is reachable from the root by a chain of roles corresponding to ρ; finally, an acyclic CQ contains m atoms that check whether $\bigcap_i \mathcal{L}(\mathcal{F}_i)$ is non-empty. We next improve this lower

bound by showing that the problem is hard already in the restricted setting where the query, the TBox, and the ABox are all fixed—just the RBox varies—and the query is arborescent. Thus, our algorithm from Chapter 9 is worst-case optimal also in KB complexity.

Theorem 10.7. *For a regular $\mathcal{ELRO}_\bot^+$ KB $\mathcal{K}$, a CQ q over $\mathcal{K}$, and a substitution π, checking $\mathcal{K} \models_{\mathcal{DL}} \pi(q)$ is* PSPACE*-hard even when*

- *the query q is fixed and arborescent;*
- *the TBox $\mathcal{T}$ is fixed and is in $\mathcal{EL}$; and*
- *the ABox $\mathcal{A}$ is fixed and contains a single unary assertion.*

Proof. The proof is by reduction from the PSPACE-hard problem of deciding whether the intersection of the languages recognised by m deterministic finite automata is non-empty [62]. Let $\mathcal{F}'_1, \dots, \mathcal{F}'_m$ be deterministic finite automata over alphabet Σ', let σ_1 and σ_2 be fresh symbols not occurring in Σ', and let $\Sigma = \Sigma' \cup \{\sigma_1, \sigma_2\}$. Then, for each $j \in [1..m]$, we let $\mathcal{F}_j = \langle Q_j, \Sigma, \delta_j, i_j, f_j \rangle$ be the deterministic FA obtained by extending $\mathcal{F}'_j$ with a transition labelled by σ_1 from the final state f'_j of $\mathcal{F}'_j$ to itself, and with a transition labelled by σ_2 from f'_j to a fresh final state f_j of $\mathcal{F}_j$. Then, $\bigcap_j \mathcal{L}(\mathcal{F}'_j) \neq \emptyset$ if and only if a word $w \in \bigcap_j \mathcal{L}(\mathcal{F}_j)$ exists such that $|w|$ is odd: given $w \in \bigcap_j \mathcal{L}(\mathcal{F}'_j)$, if $|w|$ is odd, then $|w \cdot \sigma_1 \cdot \sigma_2|$ is odd and $w \cdot \sigma_1 \cdot \sigma_2 \in \mathcal{L}(\mathcal{F}_j)$ for each $j \in [1..m]$, and if $|w|$ is even, then $|w \cdot \sigma_2|$ is odd and $w \cdot \sigma_2 \in \mathcal{L}(\mathcal{F}_j)$ for each $j \in [1..m]$. Finally, we assume w.l.o.g. that $Q_i \cap Q_j \neq \emptyset$ and $Q_i \subseteq N_{\mathfrak{R}}$ hold for each $1 \leq i < j \leq m$, and that $\Sigma \subseteq N_{\mathfrak{R}}$ as well.

Let $w = R_1 \cdots R_n$ be a word in Σ^* such that n is odd, and let $\Gamma = \Sigma \cup Q_1 \cup \ldots \cup Q_m$. Clearly, $w \in \bigcap_j \mathcal{L}(\mathcal{F}_j)$ holds if and only if a word $\rho_w \in \Gamma^*$ of the form

$$\rho_w = e_1^0 \cdots e_m^0 \cdot R_1 \cdot o_m^1 \cdots o_1^1 \cdot R_2 \cdot e_1^2 \cdots e_m^2 \cdot \cdots\cdots e_1^{n-1} \cdots e_m^{n-1} \cdot R_n \cdot o_m^n \cdots o_1^n \qquad (10.26)$$

exists such that for each $j \in [1..m]$ and each $i \in [1..n]$ we have that:

(i) i is odd implies that $o_j^i \in Q_j$ and $\delta_j(e_j^{i-1}, R_i) = o_j^i$;

(ii) i is even implies that $e_j^i \in Q_j$ and $\delta_j(o_j^{i-1}, R_i) = e_j^i$; and

(iii) $e_j^0 = i_j$ and $o_j^n = f_j$.

Now let $\mathcal{L}_O$, $\mathcal{L}_E$, $\mathcal{L}_1$, and $\mathcal{L}_2$ be the following languages.

$$\mathcal{L}_O = \{e_1 \cdots e_m \cdot R \cdot o_m \cdots o_1 \mid R \in \Sigma \text{ and } \forall j \in [1..m] : \delta_j(e_j, R) = o_j\} \tag{10.27}$$

$$\mathcal{L}_E = \{o_m \cdots o_1 \cdot R \cdot e_1 \cdots e_m \mid R \in \Sigma \text{ and } \forall j \in [1..m] : \delta_j(o_j, R) = e_j\} \tag{10.28}$$

$$\mathcal{L}_1 = (\mathcal{L}_O \cdot \Sigma)^* \cdot \mathcal{L}_O \tag{10.29}$$

$$\mathcal{L}_2 = \{i_1 \cdots i_m\} \cdot (\Sigma \cdot \mathcal{L}_E)^* \cdot \Sigma \cdot \{f_m \cdots f_1\} \tag{10.30}$$

Consider an arbitrary word $w \in \Sigma^*$ and the corresponding word $\rho_w \in \Gamma^*$. By the definition of $\mathcal{L}_1$, we have that $\rho_w \in \mathcal{L}_1$ if ρ_w is of the form (10.26) and it satisfies property (i). Similarly, by the definition of $\mathcal{L}_2$, we have that $\rho_w \in \mathcal{L}_2$ if ρ_w is of the form (10.26) and it satisfies properties (ii)–(iii). Thus, $\rho_w \in \mathcal{L}_1 \cap \mathcal{L}_2$ if and only if $w \in \bigcap_j \mathcal{L}(\mathcal{F}_j)$. For simplicity, in the rest of this proof, we will use the following equivalent formulations of $\mathcal{L}_1$ and $\mathcal{L}_2$.

$$\mathcal{L}_1 = \mathcal{L}_O \cup (\mathcal{L}_O \cdot \Sigma)^+ \cdot \mathcal{L}_O \tag{10.31}$$

$$\mathcal{L}_2 = \{i_1 \cdots i_m\} \cdot \Sigma \cdot \{f_m \cdots f_1\} \cup \{i_1 \cdots i_m\} \cdot (\Sigma \cdot \mathcal{L}_E)^+ \cdot \Sigma \cdot \{f_m \cdots f_1\} \tag{10.32}$$

We next define a KB $\mathcal{K}$ and a fixed arborescent query q such that $\mathcal{K} \models_{\mathcal{DL}} q$ if and only if $\bigcap_j \mathcal{L}(\mathcal{F}_j) \neq \emptyset$. We will present our construction in stages, and for each we will describe how it affects the universal interpretation I of $\mathcal{K}$. For simplicity, we first present $\mathcal{K}$ in which the TBox depends on Γ, and later we modify the construction to use a fixed TBox.

The TBox $\mathcal{T}$ contains axioms (10.33)–(10.34), and the ABox $\mathcal{A}$ contains only assertion (10.35). Then, for each word $\rho \in \Gamma^*$, a term w_ρ exists in I that can be reached from a_ϵ in I via a chain of roles corresponding to ρ.

$$A \sqsubseteq B \tag{10.33}$$

$$B \sqsubseteq \exists \omega . B \qquad \forall \omega \in \Gamma \tag{10.34}$$

$$A(a_\epsilon) \tag{10.35}$$

We next present an RBox $\mathcal{R}$ consisting of four parts, each encoding languages $\mathcal{L}_O$, $\mathcal{L}_E$, $\mathcal{L}_1$,

and $\mathcal{L}_2$. Our construction uses fresh roles $L_O^{R,1}, \ldots, L_O^{R,m+1}$ and $L_E^{R,0}, \ldots, L_E^{R,m}$ uniquely associated with each role $R \in \Sigma$, as well as fresh roles L_O, L_E, L_Σ, L_1, L_1', L_2, and L_2'.

The first part of $\mathcal{R}$ contains axioms (10.36)–(10.38). Please note that, for all words $\rho_1, \rho_2 \in \Gamma^*$ where ρ_1 is a prefix of ρ_2, we have $L_O(a_{\rho_1}, a_{\rho_2}) \in I$ if and only if $\rho_2 - \rho_1 \in \mathcal{L}_O$.

$$R \sqsubseteq L_O^{R,m+1} \qquad \forall R \in \Sigma \tag{10.36}$$

$$e_j \cdot L_O^{R,j+1} \cdot o_j \sqsubseteq L_O^{R,j} \qquad \forall R \in \Sigma\ \forall j \in [1..m]\ \forall e_j, o_j \in Q_j \text{ with } \delta_j(e_j, R) = o_j \tag{10.37}$$

$$L_O^{R,1} \sqsubseteq L_O \qquad \forall R \in \Sigma \tag{10.38}$$

The second part of $\mathcal{R}$ contains axioms (10.39)–(10.41). Then, for all words $\rho_1, \rho_2 \in \Gamma^*$ where ρ_1 is a prefix of ρ_2, we have $L_E(a_{\rho_1}, a_{\rho_2}) \in I$ if and only if $\rho_2 - \rho_1 \in \mathcal{L}_E$.

$$R \sqsubseteq L_E^{R,0} \qquad \forall R \in \Sigma \tag{10.39}$$

$$o_j \cdot L_E^{R,j-1} \cdot e_j \sqsubseteq L_E^{R,j} \qquad \forall R \in \Sigma\ \forall j \in [1..m]\ \forall e_j, o_j \in Q_j \text{ with } \delta_j(e_j, R) = o_j \tag{10.40}$$

$$L_E^{R,m} \sqsubseteq L_E \qquad \forall R \in \Sigma \tag{10.41}$$

The third part of $\mathcal{R}$ contains axioms (10.42)–(10.46). It should be clear that, for all words $\rho_1, \rho_2 \in \Gamma^*$ where ρ_1 is a prefix of ρ_2, we have $L_1(a_{\rho_1}, a_{\rho_2}) \in I$ if and only if $\rho_2 - \rho_1 \in \mathcal{L}_1$.

$$R \sqsubseteq L_\Sigma \qquad \forall R \in \Sigma \tag{10.42}$$

$$L_O \sqsubseteq L_1 \tag{10.43}$$

$$L_O \cdot L_\Sigma \sqsubseteq L_1' \tag{10.44}$$

$$L_1' \cdot L_1' \sqsubseteq L_1' \tag{10.45}$$

$$L_1' \cdot L_O \sqsubseteq L_1 \tag{10.46}$$

The fourth part of $\mathcal{R}$ contains axioms (10.47)–(10.50). Then, for all words $\rho_1, \rho_2 \in \Gamma^*$ where ρ_1 is a prefix of ρ_2, we have $L_2(a_{\rho_1}, a_{\rho_2}) \in I$ if and only if $\rho_2 - \rho_1 \in \mathcal{L}_2$.

$$i_1 \cdots i_m \cdot L_\Sigma \cdot f_m \cdots f_1 \sqsubseteq L_2 \tag{10.47}$$

$$L_\Sigma \cdot L_E \sqsubseteq L_2' \tag{10.48}$$

$$L'_2 \cdot L'_2 \sqsubseteq L'_2 \tag{10.49}$$

$$i_1 \cdots i_m \cdot L'_2 \cdot L_\Sigma \cdot f_m \cdots f_1 \sqsubseteq L_2 \tag{10.50}$$

Query q is given in (10.51) and it should be clear that it is arborescent. Please note that, because only individual a_ϵ satisfies concept A in the universal interpretation I of $\mathcal{K}$, each substitution that embeds q in I necessarily maps variables x_1 and x_2 to individual a_ϵ. Then, $\mathcal{K} \models_{\mathcal{DL}} q$ if and only if a word $\rho \in \Gamma^*$ exists such that $L_1(a_\epsilon, a_\rho) \in I$ and $L_2(a_\epsilon, a_\rho) \in I$, and the latter is clearly the case if and only if $\rho \in \mathcal{L}_1 \cap \mathcal{L}_2$. RBox $\mathcal{R}$ is regular and of size polynomial in the size of automata $\mathcal{F}_1, \ldots, \mathcal{F}_m$.

$$q = \exists x_1 \exists x_2 \exists y.\, A(x_1) \wedge L_1(x_1, y) \wedge A(x_2) \wedge L_2(x_2, y) \tag{10.51}$$

We next tighten this reduction to use the fixed TBox $\mathcal{T}'$ consisting of axiom (10.33) and of axioms (10.52)–(10.53), where P_0 and P_1 are fresh roles.

$$B \sqsubseteq \exists P_0.B \tag{10.52}$$

$$B \sqsubseteq \exists P_1.B \tag{10.53}$$

Let $k = \lceil \log_2 |\Gamma| \rceil$, and assume that each symbol $\omega \in \Gamma$ corresponds to a k-digit binary number $b_1 \cdots b_k$ with $b_i \in \{0, 1\}$. Then, let $\mathcal{R}'$ be $\mathcal{R}$ extended with axiom (10.54).

$$P_{b_1} \cdots P_{b_k} \sqsubseteq \omega \qquad \forall \omega \in \Gamma \text{ corresponding to } b_1 \cdots b_k \tag{10.54}$$

Finally, let $\mathcal{K}' = \mathcal{T}' \cup \mathcal{R}' \cup \mathcal{A}$, and let I' be the universal interpretation of $\mathcal{K}'$. Axioms (10.35), (10.52), and (10.53) ensure the existence of a binary tree whose edges are labelled with roles P_0 and P_1. Furthermore, axioms (10.54) ensure that, for each $\omega \in \Gamma$ and each sequence of k edges in this tree corresponding to the binary number assigned to ω, there is a 'shortcut' in the tree labelled with ω. Thus, the universal interpretation I of $\mathcal{K}$ can be homomorphically embedded into the universal interpretation I' of $\mathcal{K}'$. Finally, roles P_0 and P_1 do not occur in $\mathcal{R}$ and query q checks for existence of a domain element connected to

a_ϵ; therefore, the 'extra' edges in I' are irrelevant. Consequently, the encoding of languages $\mathcal{L}_1$ and $\mathcal{L}_2$ works in the same way as with the varying TBox $\mathcal{T}$. □

Chapter 11

Navigational Queries

The data in DL knowledge bases has a graph-like structure, where unary assertions encode properties of graph nodes and binary assertions encode graph edges. Conjunctive queries cannot express recursive properties such as reachability, and so their expressivity is often insufficient in applications that require graph navigation. As the popularity of graph databases is on the rise, a number of navigational languages for querying graph-like data have been proposed; for example, regular path queries [7] use regular expressions to express complex navigational patterns between graph vertices, and graph XPath queries [70] extend regular path queries with the converse operator, negation on regular expressions, and checking properties of vertices using Boolean combinations of concepts and existential quantifications over paths. In the DL context, the computational complexity of navigational queries has been studied for several expressive DLs and members of the DL-Lite family and the $\mathcal{EL}(\mathcal{H})$ fragment of $\mathcal{ELRO}^{+}_{\bot}$ [15, 9, 61, 11]. In order to complete the complexity landscape of this problem, in this chapter we study the problem of answering graph XPath queries over $\mathcal{ELRO}^{+}_{\bot}$ knowledge bases.

11.1 Graph XPath Queries

Graph XPath queries consist of *node expressions* and *path expressions* which are defined as specified in Table 11.1 for $A \in N_{\mathfrak{C}}$ and $R \in N_{\mathfrak{R}}$. Following Libkin et al. [70], we consider the following expression fragments.

Table 11.1: Interpreting path and node expressions in an interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$

	Syntax	Semantics
Node expression:		
atomic concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
complement	$\neg\varphi$	$\Delta^{\mathcal{I}} \setminus (\varphi)^{\mathcal{I}}$
conjunction	$\varphi_1 \wedge \varphi_2$	$(\varphi_1)^{\mathcal{I}} \cap (\varphi_2)^{\mathcal{I}}$
disjunction	$\varphi_1 \vee \varphi_2$	$(\varphi_1)^{\mathcal{I}} \cup (\varphi_2)^{\mathcal{I}}$
existential quantification	$\langle\alpha\rangle$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : \langle x, y\rangle \in \alpha^{\mathcal{I}}\}$
Path expression:		
role	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
converse	R^-	$\{\langle y, x\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \langle x, y\rangle \in R^{\mathcal{I}}\}$
concatenation	$\alpha_1 \cdot \alpha_2$	$(\alpha_1)^{\mathcal{I}} \circ (\alpha_2)^{\mathcal{I}}$
union	$\alpha_1 + \alpha_2$	$(\alpha_1)^{\mathcal{I}} \cup (\alpha_2)^{\mathcal{I}}$
Kleene's star	α^*	$(\alpha^{\mathcal{I}})^*$
negation	$\overline{\alpha}$	$(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus \alpha^{\mathcal{I}}$
node test	$\mathsf{test}(\varphi)$	$\{\langle x, x\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid x \in \varphi^{\mathcal{I}}\}$

- The *path-positive* fragment forbids path expressions of the form $\overline{\alpha}$.
- The *positive* fragment forbids path expressions of the form $\overline{\alpha}$ and node expressions of the form $\neg\varphi$.
- The *converse-free* fragment forbids path expressions of the form R^-.

A *graph XPath atom* has the form $\varphi(s)$ or $\alpha(s,t)$, for φ a node expression, α a path expression, and s and t first-order terms. A *conjunctive graph XPath query* (CGXQ) g is an expression $g = \exists\vec{y}.\, \psi(\vec{x}, \vec{y})$ where ψ is a conjunction of graph atoms over variables $\vec{x} \cup \vec{y}$; variables $\vec{x}$ are called the *answer variables* of g. If g has the form $g = \alpha(s,t)$, then g is a *graph XPath query* (GXQ). Path-positive, positive, and converse-free (C)GXQs are obtained by restricting query atoms accordingly.

The interpretation of node and path expressions in an interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ is inductively defined as shown on the right-hand side of Table 11.1. Please observe that the difference of path expressions α_1 and α_2 corresponds to $\overline{(\overline{\alpha_1} + \alpha_2)}$, whereas the intersection of α_1 and α_2 corresponds to $\overline{(\overline{\alpha_1} + \overline{\alpha_2})}$; moreover, Libkin et al. [70] define a path expression ϵ, which in our setting corresponds to $\mathsf{test}(\top_c)$. For $\mathcal{K}$ an $\mathcal{ELRO}^+_\bot$ KB, $g = \exists\vec{y}.\, \psi(\vec{x}, \vec{y})$ a CGXQ, and π a substitution with $\mathsf{dom}(\pi) = \vec{x}$ and $\mathsf{rng}(\pi) \subseteq \mathsf{ind}_{\mathcal{K}}$, $\mathcal{K} \models_{\mathcal{DL}} \pi(g)$ is defined in the obvious way, and *CGXQ answering* is the problem of checking $\mathcal{K} \models_{\mathcal{DL}} \pi(g)$.

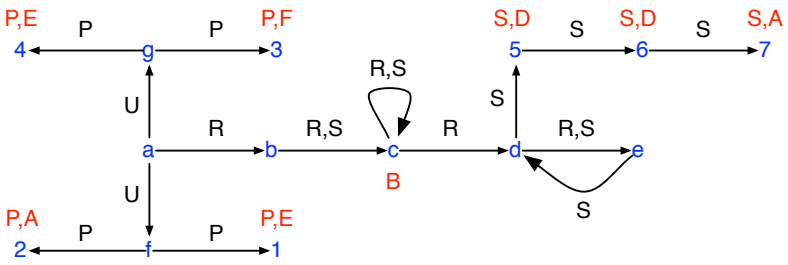

Figure 11.1: Interpretation $\mathcal{I}$

Example 11.1. *We illustrate these definitions using interpretation $\mathcal{I}$ shown in Figure 11.1; notation is as in Example 4.1 on page 39. Moreover, let α_1, α_2, and α_3 be the following path expressions.*

$$\alpha_1 = (R \cdot \mathsf{test}(\langle S^* \cdot \mathsf{test}(A \vee B)\rangle))^* \tag{11.1}$$

$$\alpha_2 = (U \cdot \mathsf{test}(\neg\langle P \cdot \mathsf{test}(A \vee B)\rangle)) \tag{11.2}$$

$$\alpha_3 = (\overline{(R \cdot S)^*}) \tag{11.3}$$

Expression α_1 is positive, and it retrieves all pairs of individuals that are connected by a path of R-edges such that, for each element occurring in the path other than the first, there exists an outgoing path of S-edges reaching a member of concept $A \sqcup B$. For example, we have $\{\langle a^{\mathcal{I}}, d^{\mathcal{I}}\rangle, \langle a^{\mathcal{I}}, e^{\mathcal{I}}\rangle\} \subseteq (\alpha_1)^{\mathcal{I}}$.

In contrast, expression α_2 is path-positive, and it retrieves all pairs of individuals that are connected by a U-edge such that no P-successor exists that is a member of concept $A \sqcup B$. For example, we have $\langle a^{\mathcal{I}}, g^{\mathcal{I}}\rangle \in (\alpha_2)^{\mathcal{I}}$, but $\langle a^{\mathcal{I}}, f^{\mathcal{I}}\rangle \notin (\alpha_2)^{\mathcal{I}}$.

Finally, expression α_3 is neither positive nor path-positive, and it retrieves all pairs of individuals that are connected by a path not *consisting of a sequence of edges described by the regular expression $(R \cdot S)^*$. For example, we have $\langle a^{\mathcal{I}}, d^{\mathcal{I}}\rangle \in (\alpha_3)^{\mathcal{I}}$, but $\langle a^{\mathcal{I}}, e^{\mathcal{I}}\rangle \notin (\alpha_3)^{\mathcal{I}}$.*

Let $g = \exists x, y, z.\ \alpha_1(x, y) \wedge \alpha_2(x, z) \wedge \alpha_3(x, y)$ be a conjunctive graph XPath query, and let $\sigma = \{x \mapsto a, y \mapsto d, z \mapsto g\}$ be a substitution. Using Figure 11.1, one can see that $\mathcal{I} \models \sigma(g)$.

As observed by Kostylev et al. [61], node expressions in graph XPath queries correspond precisely to formulas in propositional dynamic logic with negation (PDL$^{\neg}$) [48]; the satis-

Table 11.2: Encoding positive, converse-free node and path expressions using axioms

Node expressions

$\mathcal{T}_A = \{A \sqsubseteq C_A\}$	$\mathcal{R}_A = \emptyset$
$\mathcal{T}_{\varphi_1 \wedge \varphi_2} = \{C_{\varphi_1} \sqcap C_{\varphi_1} \sqsubseteq C_{\varphi_1 \wedge \varphi_2}\}$ $\cup\, \mathcal{T}_{\varphi_1} \cup \mathcal{T}_{\varphi_2}$	$\mathcal{R}_{\varphi_1 \wedge \varphi_2} = \mathcal{R}_{\varphi_1} \cup \mathcal{R}_{\varphi_2}$
$\mathcal{T}_{\varphi_1 \vee \varphi_2} = \{C_{\varphi_1} \sqsubseteq C_{\varphi_1 \vee \varphi_2}\}$ $\cup\, \{C_{\varphi_2} \sqsubseteq C_{\varphi_1 \vee \varphi_2}\}$ $\cup\, \mathcal{T}_{\varphi_1} \cup \mathcal{T}_{\varphi_2}$	$\mathcal{R}_{\varphi_1 \vee \varphi_2} = \mathcal{R}_{\varphi_1} \cup \mathcal{R}_{\varphi_2}$
$\mathcal{T}_{\langle\alpha\rangle} = \{\exists T_\alpha . \top_c \sqsubseteq C_{\langle\alpha\rangle}\} \cup \mathcal{T}_\alpha$	$\mathcal{R}_{\langle\alpha\rangle} = \mathcal{R}_\alpha$

Path expressions

$\mathcal{T}_R = \emptyset$	$\mathcal{R}_R = \{R \sqsubseteq T_R\}$
$\mathcal{T}_{\alpha_1 \cdot \alpha_2} = \mathcal{T}_{\alpha_1} \cup \mathcal{T}_{\alpha_2}$	$\mathcal{R}_{\alpha_1 \cdot \alpha_2} = \{T_{\alpha_1} \cdot T_{\alpha_2} \sqsubseteq T_{\alpha_1 \cdot \alpha_2}\}$ $\cup\, \mathcal{R}_{\alpha_1} \cup \mathcal{R}_{\alpha_2}$
$\mathcal{T}_{\alpha_1 + \alpha_2} = \mathcal{T}_{\alpha_1} \cup \mathcal{T}_{\alpha_2}$	$\mathcal{R}_{\alpha_1 + \alpha_2} = \{T_{\alpha_1} \sqsubseteq T_{\alpha_1 + \alpha_2}\}$ $\cup\, \{T_{\alpha_2} \sqsubseteq T_{\alpha_1 + \alpha_2}\}$ $\cup\, \mathcal{R}_{\alpha_1} \cup \mathcal{R}_{\alpha_2}$
$\mathcal{T}_{\alpha^*} = \mathcal{T}_\alpha$	$\mathcal{R}_{\alpha^*} = \{\epsilon \sqsubseteq T_{\alpha^*}\}$ $\cup\, \{T_\alpha \sqsubseteq T_{\alpha^*}\}$ $\cup\, \{T_{\alpha^*} \cdot T_{\alpha^*} \sqsubseteq T_{\alpha^*}\}$ $\cup\, \mathcal{R}_\alpha$
$\mathcal{T}_{\mathsf{test}(\varphi)} = \{C_\varphi \sqsubseteq \exists T_{\mathsf{test}(\varphi)} . \mathsf{Self}\} \cup \mathcal{T}_\varphi$	$\mathcal{R}_{\mathsf{test}(\varphi)} = \mathcal{R}_\varphi$

fiability problem for PDL$^\neg$ is undecidable [47], so answering GXQs under DL constraints is undecidable. Decidability results have been recently obtained for path-positive and positive queries over *DL-Lite* and $\mathcal{EL}$ knowledge bases [61, 11]. In addition, Kostylev et al. [61] proved that, for all DLs, answering path-positive, converse-free GXQs is coNP-hard in data-complexity. Finally, Bienvenu et al. [11] proved that answering positive GXQs over $\mathcal{EL}$ knowledge bases is ExpTime-complete. Thus, in the rest of this section we focus on positive, converse-free graph XPath queries.

11.2 Positive, Converse-Free CGXQ over $\mathcal{ELRO}^+_\bot$ KBs

In the rest of this section, we fix an $\mathcal{ELRO}^+_\bot$ KB $\mathcal{K} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$ such that $\mathcal{R}$ is regular. We next show that, given a positive, converse-free CGXQ g and a substitution π, one can construct in polynomial time a regular $\mathcal{ELRO}^+_\bot$ KB $\mathcal{K}'$ and a CQ q' such that $\mathcal{K} \models_{\mathcal{DL}} \pi(g)$

if and only if $\mathcal{K}' \models_{\mathcal{DL}} \pi(q')$. Our construction of $\mathcal{K}'$ combines various expressive features of $\mathcal{ELRO}^+_\bot$: role inclusions and reflexive roles encode the path expressions of g in the RBox, and self-restrictions encode the node expressions of g in the TBox.

Proposition 11.1. *Given a positive, converse-free CGXQ $g = \exists\vec{y}.\ \psi(\vec{x}, \vec{y})$ one can compute in time polynomial in $|\mathcal{K}| + |g|$ an $\mathcal{ELRO}^+_\bot$ KB $\mathcal{K}'$ and a CQ $q' = \exists\vec{y}.\ \psi'(\vec{x}, \vec{y})$ such that*

- *the RBox of $\mathcal{K}'$ is regular,*
- *g and q have equally many atoms, and*
- *for each substitution π, $\mathcal{K} \models_{\mathcal{DL}} \pi(g)$ if and only if $\mathcal{K}' \models_{\mathcal{DL}} \pi(q')$.*

Proof. Let $g = \exists\vec{y}.\ \psi(\vec{x}, \vec{y})$ be a positive, converse-free CGXQ, and let π be a substitution mapping $\vec{x}$ to $N_\mathfrak{I}$. For each positive node expression φ, let C_φ be a fresh atomic concept uniquely associated with φ and, for each positive, converse-free path expression α, let T_α be a fresh role uniquely associated with α. By structural induction, we associate with each φ (resp. α) a TBox $\mathcal{T}_\varphi$ and an RBox $\mathcal{R}_\varphi$ (resp. a TBox $\mathcal{T}_\alpha$ and an RBox $\mathcal{R}_\alpha$) as shown in Table 11.2. Then, let $\mathcal{K}' = \mathcal{T} \cup \mathcal{T}' \cup \mathcal{R} \cup \mathcal{R}' \cup \mathcal{A}$ where $\mathcal{T}'$ and $\mathcal{R}'$ are as follows.

$$\mathcal{T}' = \bigcup_{\varphi(s)\in\psi} \mathcal{T}_\varphi \cup \bigcup_{\alpha(s,t)\in\psi} \mathcal{T}_\alpha \qquad \mathcal{R}' = \bigcup_{\varphi(s)\in\psi} \mathcal{R}_\varphi \cup \bigcup_{\alpha(s,t)\in\psi} \mathcal{R}_\alpha$$

Now let $q' = \exists\vec{y}.\ \psi'(\vec{x}, \vec{y})$ be the Boolean CQ where ψ' contains $C_\varphi(s)$ for each atom $\varphi(s) \in \psi$ and $T_\alpha(s, t)$ for each atom $\alpha(s, t) \in \psi$. Clearly, g and q' have the same terms and number of atoms. Finally, both q' and $\mathcal{K}'$ can be computed in polynomial time in the input size, and the RBox of $\mathcal{K}'$ is clearly regular. We next show that $\mathcal{K}' \not\models_{\mathcal{DL}} \pi(q')$ if and only if $\mathcal{K} \not\models_{\mathcal{DL}} \pi(g)$.

($\Rightarrow$) Assume that $\mathcal{K}' \not\models_{\mathcal{DL}} \pi(q')$, so an interpretation $\mathcal{I}$ exists such that $\mathcal{I} \models \mathcal{K}'$ and $\mathcal{I} \not\models \pi(q')$. Since each axiom of $\mathcal{K}$ is also an axiom of $\mathcal{K}'$, we have that $\mathcal{I} \models \mathcal{K}$. Furthermore, for each positive node expression φ and positive path expression α, we have $\varphi^\mathcal{I} \subseteq (C_\varphi)^\mathcal{I}$ and $\alpha^\mathcal{I} \subseteq (T_\alpha)^\mathcal{I}$. We prove this claim by simultaneous induction on the structure of node and path expressions.

Base case. Let φ be an arbitrary node expression of the form $\varphi = A$ and let α be an arbitrary path expression of the form $\alpha = R$. Since $A \sqsubseteq C_A \in \mathcal{T}'$, $R \sqsubseteq T_R \in \mathcal{R}'$, and $\mathcal{I}$ is a model of $\mathcal{K}'$, the claim easily follows.

Inductive step. For the inductive step, we distinguish two cases.

First, consider an arbitrary node expression φ such that the property holds for all node and path expressions occurring in φ. Then let x be an arbitrary element of $\Delta^{\mathcal{I}}$ and assume that $x \in \varphi^{\mathcal{I}}$; we show that $x \in C_{\varphi}^{\mathcal{I}}$ by considering the various forms that φ can take.

- $\varphi = \varphi_1 \wedge \varphi_2$. Since $x \in \varphi^{\mathcal{I}}$, we have $x \in \varphi_1^{\mathcal{I}}$ and $x \in \varphi_2^{\mathcal{I}}$. By the inductive hypothesis, we have $x \in C_{\varphi_1}^{\mathcal{I}}$ and $x \in C_{\varphi_2}^{\mathcal{I}}$. By the definition of $\mathcal{T}'$, we have $C_{\varphi_1} \sqcap C_{\varphi_2} \sqsubseteq C_{\varphi} \in \mathcal{T}'$. Since $\mathcal{I}$ is a model of $\mathcal{T}'$, we have $x \in C_{\varphi}^{\mathcal{I}}$, as required.

- $\varphi = \varphi_1 \vee \varphi_2$. The proof for this case is similar to the one above.

- $\varphi = \langle \alpha \rangle$. Since $x \in \varphi^{\mathcal{I}}$, there exists $y \in \Delta^{\mathcal{I}}$ with $\langle x, y \rangle \in \alpha^{\mathcal{I}}$. By the inductive hypothesis, we have $\langle x, y \rangle \in T_{\alpha}^{\mathcal{I}}$. By the definition of $\mathcal{T}'$, we have $\exists T_{\alpha}.\top_c \sqsubseteq C_{\varphi} \in \mathcal{T}'$. Since $\mathcal{I}$ is a model of $\mathcal{T}'$, we have $x \in C_{\varphi}^{\mathcal{I}}$, as required.

Second, consider an arbitrary path expression α such that the property holds for all node and path expressions occurring in α. Then let x and y be elements of $\Delta^{\mathcal{I}}$ and assume that $\langle x, y \rangle \in \alpha^{\mathcal{I}}$; we show that $\langle x, y \rangle \in T_{\alpha}^{\mathcal{I}}$ by considering the forms that α can take.

- $\alpha = \alpha_1 \cdot \alpha_2$. Since $\langle x, y \rangle \in \alpha^{\mathcal{I}}$, there exists $z \in \Delta^{\mathcal{I}}$ such that $\langle x, z \rangle \in \alpha_1^{\mathcal{I}}$ and $\langle z, y \rangle \in \alpha_2^{\mathcal{I}}$. By the inductive hypothesis, we have $\langle x, z \rangle \in T_{\alpha_1}^{\mathcal{I}}$ and $\langle z, y \rangle \in T_{\alpha_2}^{\mathcal{I}}$. Moreover, by the definition of $\mathcal{R}'$, we have $T_{\alpha_1} \cdot T_{\alpha_2} \sqsubseteq T_{\alpha} \in \mathcal{R}'$. Since $\mathcal{I}$ is a model of $\mathcal{R}'$, we have $\langle x, y \rangle \in T_{\alpha}^{\mathcal{I}}$, as required.

- $\alpha = \alpha_1 + \alpha_2$. Given that $\langle x, y \rangle \in \alpha^{\mathcal{I}}$, we have that $\langle x, y \rangle \in \alpha_1^{\mathcal{I}}$ or $\langle x, y \rangle \in \alpha_2^{\mathcal{I}}$. By the inductive hypothesis, we have $\langle x, y \rangle \in T_{\alpha_1}^{\mathcal{I}}$ or $\langle x, y \rangle \in T_{\alpha_2}^{\mathcal{I}}$. Moreover, by the definition of $\mathcal{R}'$, we have that $\{T_{\alpha_1} \sqsubseteq T_{\alpha}, T_{\alpha_2} \sqsubseteq T_{\alpha}\} \subseteq \mathcal{R}'$; and because $\mathcal{I}$ is a model of $\mathcal{R}'$, we have $\langle x, y \rangle \in T_{\alpha}^{\mathcal{I}}$.

- $\alpha = \alpha_1^*$. First, consider the case in which $x = y$. By the definition of $\mathcal{R}'$, we have $\epsilon \sqsubseteq T_{\alpha} \in \mathcal{R}'$. Since $\mathcal{I}$ is a model of $\mathcal{R}'$, we have $\langle x, y \rangle \in T_{\alpha}^{\mathcal{I}}$, as required. Otherwise,

consider the case in which $x \neq y$. Since $\langle x, y\rangle \in \alpha^{\mathcal{I}}$, elements $x_0, \ldots, x_n$ with $x_0 = x$ and $x_n = y$ exist in $\Delta^{\mathcal{I}}$ such that $n > 0$ and $\langle x_{i-1}, x_i\rangle \in \alpha_1^{\mathcal{I}}$ for each $i \in [1..n]$. By the inductive hypothesis, for each $i \in [1..n]$, we have $\langle x_{i-1}, x_i\rangle \in T_{\alpha_1}^{\mathcal{I}}$. By the definition of $\mathcal{R}'$, we have $T_{\alpha_1} \cdot T_{\alpha_1} \sqsubseteq T_\alpha \in \mathcal{R}'$. Since $\mathcal{I}$ is a model of $\mathcal{R}'$, we have $\langle x, y\rangle \in T_\alpha$.

- $\alpha = \mathsf{test}(\varphi)$. It follows that $x = y$ and that $x \in \varphi^{\mathcal{I}}$. By the inductive hypothesis, we have $x \in C_\varphi^{\mathcal{I}}$. By the definition of $\mathcal{T}'$, we have $C_\varphi \sqsubseteq \exists T_\alpha.\mathsf{Self} \in \mathcal{T}'$. Since $\mathcal{I}$ is a model of $\mathcal{T}'$, we have $\langle x, y\rangle \in T_\alpha^{\mathcal{I}}$, as required.

Since node and path expressions in g are positive, we have $\mathcal{I} \not\models \pi(q')$ implies $\mathcal{I} \not\models \pi(g)$.

($\Leftarrow$) Assume that $\mathcal{K} \not\models_{\mathcal{DL}} \pi(g)$, so an interpretation $\mathcal{I}$ exists such that $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \not\models \pi(g)$. Let $\mathcal{I}'$ be the interpretation obtained by extending $\mathcal{I}$ to the fresh concepts and roles occurring in $\mathcal{K}'$ as follows.

$$(C_\varphi)^{\mathcal{I}'} = \varphi^{\mathcal{I}'} \qquad\qquad (T_\alpha)^{\mathcal{I}'} = \alpha^{\mathcal{I}'}$$

By the definition of $\mathcal{K}'$, it is straightforward to see that $\mathcal{I}' \models \mathcal{K}'$; furthermore, by the definition of q', it is straightforward to see that $\mathcal{I}' \not\models \pi(q')$, as required. □

Next, we establish the complexity of answering positive, converse-free (C)GXQs over $\mathcal{ELRO}_\bot^+$ knowledge bases.

Theorem 11.2. *For a positive, converse-free CGXQ g and a substitution π, checking whether $\mathcal{K} \models_{\mathcal{DL}} \pi(g)$ is* PTime*-complete in data complexity and* PSpace*-complete in combined and KB complexities. If g is a GXQ, then checking whether $\mathcal{K} \models_{\mathcal{DL}} \pi(g)$ is* PTime*-complete in combined complexity.*

Proof. Note that the hardness in data complexity of positive, converse-free (C)GXQs follows from the PTime-hardness of instance checking in $\mathcal{EL}$ [13].

For positive, converse-free GXQs, hardness in combined complexity is inherited from the PTime-hardness of TBox reasoning in $\mathcal{EL}$ [4]. For the matching upper bounds, Proposition 11.1 allows us to reduce GXQ answering to checking entailments of the form $\mathcal{K}' \models_{\mathcal{DL}} \pi(q')$ where $\pi(q')$ is a BCQ containing only one ground binary atom—that is, $\pi(q')$ is of the form

$R(a, b)$. Clearly, we have that $\mathcal{K}' \models_{\mathcal{DL}} R(a, b)$ if and only if $\mathcal{K}' \models_{\mathcal{DL}} \{a\} \sqsubseteq \exists R.\{b\}$. Since we can check in polynomial time the entailment of concept inclusions over $\mathcal{K}'$, the claim easily follow.

For positive, converse-free CGXQs, hardness in combined and KB complexities is given by Theorem 9.8, and the upper bounds follow from Theorem 9.7 and Proposition 11.1. □

Chapter 12

The OWL 2 DL Regularity Restriction

In this thesis, we presented novel complexity results and algorithms for answering expressive queries over knowledge bases formulated in several important fragments of OWL 2 EL with complex role inclusions. Complex role inclusions are closely related to context-free grammars and are a known source of undecidability for CQ answering in $\mathcal{EL}$ variants, so we have considered only role inclusions that satisfy the syntactic regularity restriction by Horrocks and Sattler [51]. This restriction ensures that each language $\mathcal{L}(R)$ induced by a knowledge base can be recognised by a finite automaton and is therefore regular. Hence, the undecidability of CQ answering does not apply to KBs with regular role inclusions.

Complex role inclusions are also known to cause the undecidability of standard reasoning tasks in expressive DLs such as $\mathcal{SROIQ}$—the DL logically underpinning the DL profile of OWL 2. For this reason, OWL 2 DL incorporates an extended version of the regularity restriction by Horrocks and Sattler [51] into its definition; moreover, this extended regularity restriction is inherited by all the profiles of OWL 2, including OWL 2 EL.

While designing the regularity restriction for OWL 2 DL, Motik et al. [78] attempted to address two known issues with the syntactic regularity restriction by Horrocks and Sattler [51]. First, each knowledge base that satisfies the restriction by Horrocks and Sattler [51] cannot contain complex role inclusions of the form $R_1 \cdots R_n \sqsubseteq R^-$, which express the com-

positional properties of the inverse of role R. Second, the absence of complex role inclusions in a knowledge base is not sufficient for regularity, although each language $\mathcal{L}(R)$ induced by the knowledge base is regular. Unfortunately, we show in Section 12.1 that the extended regularity condition by Motik et al. [78] is flawed: it does not ensure that each language $\mathcal{L}(R)$ induced by an OWL 2 knowledge base is regular.

Hence, in Section 12.2 we propose a revised syntactic condition on role inclusions for OWL 2 DL. Like the original restriction by Horrocks and Sattler [51], our syntactic condition can be checked in polynomial time. In addition, our condition supports role inclusions of the form $R_1 \cdots R_n \sqsubseteq R^-$ for the specification of the compositional properties of inverse roles, and extends to knowledge bases that do not contain complex role inclusions. Finally, we show that each DL knowledge base $\mathcal{K}$ whose RBox component satisfies our revised restriction can be transformed in polynomial time into a knowledge base $\mathcal{K}'$ that satisfies the condition by Horrocks and Sattler [51] without affecting the answers to CQs. Therefore, our restriction correctly ensures that each language $\mathcal{L}(R)$ induced by $\mathcal{K}$ is regular; moreover, if $\mathcal{K}$ is formulated in a fragment of OWL 2 EL, we can answer queries over $\mathcal{K}$ using the knowledge base $\mathcal{K}'$ and the CQ answering algorithms presented in this thesis.

12.1 The Existing Regularity Restrictions

All the profiles of OWL 2, apart from the EL profile, support inverse roles. In order to present the syntactic conditions on role inclusions with inverse roles, we next generalise our definitions of roles, role inclusions, and RBoxes to accommodate for inverse role.

Each element of $N_{\mathfrak{R}}$ is an *atomic role*; furthermore, for each $R \in N_{\mathfrak{R}} \setminus \{\top_r, \bot_r\}$, expression R^- denotes the inverse of R. Then, a *role* is either an atomic role or an inverse role. A *role chain* ρ is a possibly empty sequence of roles. Function $\mathsf{inv}(\cdot)$ maps each role chain to its inverse as follows, where R is a role and $R_1 \cdots R_n$ is a non-empty role chain.

$$\mathsf{inv}(\top_r) = \top_r \qquad \mathsf{inv}(\bot_r) = \bot_r$$

$$\mathsf{inv}(R) = R^- \qquad \mathsf{inv}(R^-) = R$$

$$\mathsf{inv}(\epsilon) = \epsilon \qquad \mathsf{inv}(R_1 \cdots R_n) = \mathsf{inv}(R_n) \cdots \mathsf{inv}(R_1)$$

For $\mathcal{I}$ a DL interpretation, the inverse role R^- is interpreted as

$$(R^-)^{\mathcal{I}} = \{\langle y, x\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \langle x, y\rangle \in R^{\mathcal{I}}\}.$$

An *RBox* $\mathcal{R}$ is a finite set of role inclusions of the form $\rho \sqsubseteq R$, where ρ is a role chain and R is a role; $\mathsf{rol}_{\mathcal{R}}$ is the smallest set that contains $\top_r$ and $\bot_r$, and contains R and $\mathsf{inv}(R)$ for each role R occurring in $\mathcal{R}$. Such an RBox $\mathcal{R}$ is a *role hierarchy* if each $\rho \sqsubseteq R \in \mathcal{R}$ satisfies $|\rho| \leq 1$ or $\rho = R \cdot R$. A role $R \in \mathsf{rol}_{\mathcal{R}}$ is *composite* in $\mathcal{R}$ if a role chain ρ with $|\rho| > 1$ exists such that either $\rho \sqsubseteq R \in \mathcal{R}$ and $\rho \neq R \cdot R$, or $\rho \sqsubseteq R^- \in \mathcal{R}$ and $\rho \neq R^- \cdot R^-$. The *subrole relation* $\sqsubseteq^*_{\mathcal{R}}$ for $\mathcal{R}$ is the smallest reflexive and transitive relation on $\mathsf{rol}_{\mathcal{R}}$ such that

- $S \sqsubseteq^*_{\mathcal{R}} R$ and $\mathsf{inv}(S) \sqsubseteq^*_{\mathcal{R}} \mathsf{inv}(R)$ for each role inclusion $S \sqsubseteq R \in \mathcal{R}$, and
- $S \sqsubseteq^*_{\mathcal{R}} \top_r$ and $\mathsf{inv}(S) \sqsubseteq \top_r$ for each role $S \in \mathsf{rol}_{\mathcal{R}}$

In Definition 12.1, we present the regularity restriction by Horrocks and Sattler [51], which is based on checking the existence of a so-called *regular order* on the roles occurring in the RBox. Horrocks and Sattler [51] showed that, for each role R occurring in a regular RBox $\mathcal{R}$, using time exponential in $|\mathcal{R}|$ one can construct a finite automaton $\mathcal{F}_R$ such that $\mathcal{L}(\mathcal{F}_R) = \mathcal{L}(R)$. When the RBox $\mathcal{R}$ does not contain inverse roles, the definition of regular RBox given below is equivalent to the one presented in Section 3.2 for $\mathcal{ELRO}^+_\bot$ RBoxes.

Definition 12.1 ([51])**.** *A strict partial order $\prec_r$ on the set of roles is a* regular order *if, for all roles S and R, $S \prec_r R$ if and only if $\mathsf{inv}(S) \prec_r R$. Then, an RBox $\mathcal{R}$ is* regular *if a regular order $\prec_r$ exists such that each axiom $\rho \sqsubseteq R \in \mathcal{R}$ with $R \neq \top_r$ satisfies $R \in N_{\mathfrak{R}}$ and $\rho \sqsubseteq R$ is of the form*

(inv) $\mathsf{inv}(R) \sqsubseteq R$, *or*

(t1) $\epsilon \sqsubseteq R$, *or*

(t2) $R \cdot R \sqsubseteq R$, *or*

(t3) $R_1 \cdots R_n \cdot R \sqsubseteq R$ *where* $n \geq 0$ *and* $R_i \prec_r R$ *for each* $i \in [1..n]$, *or*

(t4) $R_1 \cdots R_n \sqsubseteq R$ *where* $n \geq 1$ *and* $R_i \prec_r R$ *for each* $i \in [1..n]$, *or*

(t5) $R \cdot R_1 \cdots R_n \sqsubseteq R$ *where* $n \geq 0$ *and* $R_i \prec_r R$ *for each* $i \in [1..n]$.

In Example 12.1 we show that the absence of complex role inclusions is not a sufficient precondition for regularity.

Example 12.1. *Consider the RBox* $\mathcal{R} = \{S \sqsubseteq R, R \sqsubseteq S\}$. *Because* $S \sqsubseteq R \in \mathcal{R}$ *and* $R \sqsubseteq S \in \mathcal{R}$, *we must have that* $S \prec_r R$ *and* $R \prec_r S$; *but then,* $\prec_r$ *cannot be a regular order.*

Even though $\mathcal{R}$ *is not (syntactically) regular, languages* $\mathcal{L}(S)$ *and* $\mathcal{L}(R)$ *are finite, whereas* $\mathcal{L}(\bot_r)$ *is empty and* $\mathcal{L}(\top_r)$ *contains each role chain over the alphabet* $\mathsf{rol}_\mathcal{R}$. *Therefore, each language* $\mathcal{L}(R)$ *induced by* $\mathcal{R}$ *is regular.*

We next present the definition of so-called *OWL regular* RBoxes as defined in the OWL 2 specification by Motik et al. [78].

Definition 12.2 ([78])**.** *An RBox* $\mathcal{R}$ *is* OWL regular *if a regular order* $\prec_r$ *exists such that*

- $R \not\sqsubseteq^*_\mathcal{R} S$ *for each* $S \prec_r R$, *and*
- *each axiom* $\rho \sqsubseteq R \in \mathcal{R}$ *with* $R \neq \top_r$ *and* $|\rho| > 1$ *is of the form*

  (t2) $R \cdot R \sqsubseteq R$, *or*

  (t3) $R_1 \cdots R_n \cdot R \sqsubseteq R$ *where* $R_i \prec_r R$ *for each* $i \in [1..n]$, *or*

  (t4) $R_1 \cdots R_n \sqsubseteq R$ *where* $R_i \prec_r R$ *for each* $i \in [1..n]$, *or*

  (t5) $R \cdot R_1 \cdots R_n \sqsubseteq R$ *where* $R_i \prec_r R$ *for each* $i \in [1..n]$.

Each role hierarchy $\mathcal{R}$ is trivially OWL regular since all the properties in Definition 12.2 are satisfied by the empty regular order. However, the following example shows that an OWL regular RBox exists that induces a non-regular language.

Example 12.2. *Let* $\mathcal{R}$ *be the RBox containing axioms* (12.1)–(12.4).

$$R_a \cdot S \cdot R_b \sqsubseteq R \tag{12.1}$$

$$R' \sqsubseteq S \tag{12.2}$$

$$R'_a \cdot S' \cdot R'_b \sqsubseteq R' \tag{12.3}$$

$$R \sqsubseteq S' \tag{12.4}$$

Moreover, let $\prec_r$ be the following regular order.

$$R_a \prec_r R \qquad R_a^- \prec_r R$$
$$S \prec_r R \qquad S^- \prec_r R$$
$$R_b \prec_r R \qquad R_b^- \prec_r R$$
$$R'_a \prec_r R' \qquad R'^-_a \prec_r R'$$
$$S' \prec_r R' \qquad S'^- \prec_r R'$$
$$R'_b \prec_r R' \qquad R'^-_b \prec_r R'$$

Note that, for each $P \prec_r T$, we have that $T \not\sqsubseteq^*_{\mathcal{R}} P$. Then, using Definition 12.2, one can easily check that $\mathcal{R}$ is OWL regular. However, the RBox $\mathcal{R}$ induces the language $\mathcal{L}(R)$ that contains each word $\rho_a \cdot X \cdot \rho_b$ such that $X \in \{S, S', R', R\}$, ρ_a is a non-empty sequence of alternating R_a and R'_a, and ρ_b is a non-empty a sequence of alternating R_b and R'_b such that $|\rho_a| = |\rho_b|$. Since finite automata have finite memory [50], the language $\mathcal{L}(R)$ cannot be recognised by a finite automaton and so $\mathcal{L}(R)$ is not regular.

In contrast, the regular order $\prec_r$ cannot be used to show that $\mathcal{R}$ is regular. Indeed, $\mathcal{R}$ contains $R' \sqsubseteq S$ and $R \sqsubseteq S'$, but $R' \not\prec_r S$ and $R \not\prec_r S'$. Note that, if we add $R' \prec_r S$ and $R \prec_r S'$, then the resulting partial order is reflexive and thus not a regular order.

Hence, the decidability proof by Horrocks et al. [53] for consistency checking in the description logic underpinning OWL 2 DL does not apply to knowledge bases with OWL regular RBoxes. Similarly, our decidability proof and the decidability proof by Ortiz et al. [83] for CQ answering over OWL 2 EL knowledge bases do not apply in case of OWL regular RBoxes. Moreover, the ability to encode non-regular languages in a knowledge base brings the decidability of these two problems into question.

12.2 A Revised Regularity Restriction for OWL 2 DL

Example 12.2 shows that the OWL regularity restriction overlooks the interplay between the simple role inclusions of the form $S \sqsubseteq R$ and the complex role inclusions of the form $R_1 \cdots R_n \sqsubseteq R$ occurring in an RBox $\mathcal{R}$. We next solve this issue and present our revised

syntactic regularity restriction for OWL 2 DL. Unlike the existing conditions which are non-constructive because they require checking the existence of a regular order, our condition is based on checking an acyclicity condition on a graph associated with the RBox $\mathcal{R}$; thus, our restriction can be implemented more directly. More specifically, in Definition 12.3 we use the role inclusions in $\mathcal{R}$ to construct a directed graph $G_{\mathcal{R}}$; and then, we determine whether $\mathcal{R}$ satisfies our *weak regularity* restriction by checking that (i) $G_{\mathcal{R}}$ does not contain cycles involving composite roles, and (ii) all the role inclusions in $\mathcal{R}$ are in a normal form similar to the one specified by Horrocks and Sattler [51] and Motik et al. [78].

Definition 12.3. *For an RBox $\mathcal{R}$, let $G_{\mathcal{R}} = \langle \mathsf{rol}_{\mathcal{R}}, E \rangle$ be the directed graph where E is the smallest set of edges such that $\{\langle R_i, R\rangle, \langle \mathsf{inv}(R_i), R\rangle\} \subseteq E$ for each axiom $R_1 \cdots R_n \sqsubseteq R \in \mathcal{R}$ with $R \neq \top_r$, and each $i \in [1..n]$ with $R_i \neq R$ and $\mathsf{inv}(R_i) \neq R$. Then, RBox $\mathcal{R}$ is* weakly regular *if each composite role $R \in \mathsf{rol}_{\mathcal{R}}$ is not reachable from itself in $G_{\mathcal{R}}$ and each axiom $\rho \sqsubseteq R \in R$ with $R \neq \top_r$ is of the form*

(inv) $\mathsf{inv}(R) \sqsubseteq R$, *or*

(t1) $\epsilon \sqsubseteq R$, *or*

(t2) $R \cdot R \sqsubseteq R$, *or*

(t3) $R_1 \cdots R_n \cdot R \sqsubseteq R$ *where* $n \geq 0$, *and* $R_i \neq R$ *and* $R_i \neq \mathsf{inv}(R_i)$ *for each* $i \in [1..n]$, *or*

(t4) $R_1 \cdots R_n \sqsubseteq R$ *where* $n \geq 1$, *and* $R_i \neq R$ *and* $R_i \neq \mathsf{inv}(R_i)$ *for each* $i \in [1..n]$, *or*

(t5) $R \cdot R_1 \cdots R_n \sqsubseteq R$ *where* $n \geq 0$, *and* $R_i \neq R$ *and* $R_i \neq \mathsf{inv}(R_i)$ *for each* $i \in [1..n]$.

Please note that each role hierarchy $\mathcal{R}$ is weakly regular, because role hierarchies do not contain composite roles. In the following examples, we illustrate weakly regular RBoxes.

Example 12.3. *Consider the RBox $\mathcal{R}$ from Example 12.1. No roles are composite in $\mathcal{R}$ since $\mathcal{R}$ does not contain complex role inclusions. The left part of Figure 12.1 shows the graph $G_{\mathcal{R}}$ associated with $\mathcal{R}$. Although the graph is cyclic, the RBox $\mathcal{R}$ is weakly regular.*

Example 12.4. *Consider the RBox $\mathcal{R}$ from Example 12.2. Roles R and R' are composite in $\mathcal{R}$ due to axioms (12.1) and (12.3), respectively. The right part of Figure 12.1 shows the*

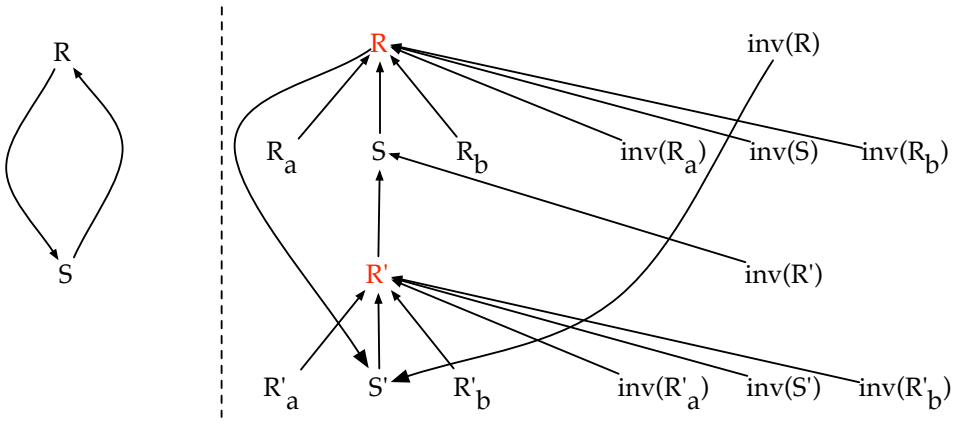

Figure 12.1: The graphs associated with the RBoxes in Examples 12.1 and 12.2, respectively

graph $G_{\mathcal{R}}$ associated with $\mathcal{R}$. Roles that are composite in $\mathcal{R}$ are shown in red. As one can see, both R and R' are reachable from themselves in $G_{\mathcal{R}}$, and so $\mathcal{R}$ is not weakly regular.

Theorem 12.4 shows that each DL knowledge base $\mathcal{K}$ with a weakly-regular RBox can be 'regularised' in polynomial time without affecting the answers to CQs. This result thus shows that weakly regular RBoxes induce only regular languages; moreover, it shows that our CQ answering algorithms from Chapters 6 and 9 can be used to answer conjunctive queries over $\mathcal{ELHO}_{\bot}^{+}$ and $\mathcal{ELRO}_{\bot}^{+}$ knowledge bases with weakly regular RBoxes as well.

Theorem 12.4. *For each first-order expressible DL knowledge base $\mathcal{K}$ with a weakly regular RBox and each conjunctive query q over $\mathcal{K}$, one can compute in polynomial time a knowledge base $\mathcal{K}'$ and a conjunctive query q' over $\mathcal{K}'$, such that*

- *the RBox of $\mathcal{K}'$ is regular, and*
- *For each substitution π, $\mathcal{K} \models_{\mathcal{DL}} \pi(q)$ if and only if $\mathcal{K}' \models_{\mathcal{DL}} \pi(q')$.*

Proof. Let $\mathcal{K} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$ be an arbitrary knowledge base with a weakly regular RBox that can be equivalently expressed as first-order theory; furthermore, let $G_{\mathcal{R}} = \langle \mathsf{rol}_{\mathcal{R}}, E \rangle$ be the graph associated with $\mathcal{R}$. Let q be a conjunctive query over $\mathcal{K}$ and let π be an arbitrary substitution.

In the rest of this proof, we consider the case in which $\top_r \not\sqsubseteq^{*}_{\mathcal{R}} \bot_r$ since otherwise the theorem immediately holds for the KB $\mathcal{K}' = \{\top_r \sqsubseteq \bot_r\}$ and the query q' defined as $q' = q$.

We first compute KB $\mathcal{K}_1 = \mathcal{T} \cup \mathcal{R}_1 \cup \mathcal{A}$ such that $\mathcal{R}_1$ is weakly regular and each role inclusion $\rho \sqsubseteq R \in \mathcal{R}_1$ satisfies $R \in N_{\mathfrak{R}}$. To this end, let $\mathcal{R}_1$ be the result of replacing each role inclusion $\rho \sqsubseteq R \in \mathcal{R}$ such that R is an inverse role with $\mathsf{inv}(\rho) \sqsubseteq \mathsf{inv}(R)$. The resulting KB $\mathcal{K}_1$ can be computed in polynomial time. Furthermore, for each interpretation $\mathcal{I}$, we have that $\mathcal{I}$ satisfies $\rho \sqsubseteq R$ if and only if $\mathcal{I}$ satisfies $\mathsf{inv}(\rho) \sqsubseteq \mathsf{inv}(R)$, and so so $\mathcal{K} \models_{\mathcal{DL}} \pi(q)$ if and only if $\mathcal{K}_1 \models_{\mathcal{DL}} \pi(q)$. We are left to show that $\mathcal{R}_1$ is weakly regular.

To this end, let $G_{\mathcal{R}_1} = \langle \mathsf{rol}_{\mathcal{R}_1}, E_1 \rangle$ be the graph associated with $\mathcal{R}_1$. We first show that for each non-empty path $R_0, \ldots, R_n$ in $G_{\mathcal{R}_1}$, a path $P_0, \ldots, P_n$ in $G_{\mathcal{R}}$ exists such that $P_i \in \{R_i, \mathsf{inv}(R_i)\}$ for each $i \in [0..n]$. The proof is by induction on the length of the path $n \in \mathbb{N}^+$. For the base case, consider an arbitrary edge $\langle R_0, R_1 \rangle$ of $G_{\mathcal{R}_1}$. By the construction of $\mathcal{R}_1$, we have that either $\langle R_0, R_1 \rangle$ is an edge of $G_{\mathcal{R}}$, or $\langle \mathsf{inv}(R_0), \mathsf{inv}(R_1) \rangle$ is an edge of $G_{\mathcal{R}}$, so the property holds in the base case. For the inductive step, consider an arbitrary $n \in \mathbb{N}^+$ and assume that the property holds for each path of length n; we show that the same holds for all paths of length $n+1$. To this end, consider an arbitrary path $R_0, \ldots, R_{n+1}$ in $G_{\mathcal{R}_1}$ of length $n+1$. By the inductive hypothesis, there exists a path $P_0, \ldots, P_n$ in $G_{\mathcal{R}}$ such that $P_i \in \{R_i, \mathsf{inv}(R_i)\}$ for each $i \in [0..n]$. Next, consider the form of the edge $\langle R_n, R_{n+1} \rangle \in E_1$. If $\langle R_n, R_{n+1} \rangle$ occurs in $G_{\mathcal{R}}$, then we have that $\langle \mathsf{inv}(R_n), R_{n+1} \rangle$ also occurs in $G_{\mathcal{R}}$, and so $P_0, \ldots, P_n, R_{n+1}$ is the required path. Otherwise, if $\langle R_n, R_{n+1} \rangle$ does not occur in $G_{\mathcal{R}}$, then $\langle \mathsf{inv}(R_n), \mathsf{inv}(R_{n+1}) \rangle$ and $\langle R_n, \mathsf{inv}(R_{n+1}) \rangle$ must occur in $G_{\mathcal{R}}$, by the definition of $\mathcal{R}_1$, and so $P_0, \ldots, P_n, \mathsf{inv}(R_{n+1})$ is the required path.

We are ready to show that $\mathcal{R}_1$ is weakly regular. Assume the opposite; hence, roles $R_0, \ldots, R_n$ exist in $\mathsf{rol}_{\mathcal{R}_1}$ such that $\langle R_{i-1}, R_i \rangle \in E_1$ for each $i \in [1..n]$, and $R_0 = R_n$ and R_n is composite in $\mathcal{R}_1$. By the above property, a path $P_0, \ldots, P_n$ exist in $G_{\mathcal{R}}$ such that $P_i \in \{R_i, \mathsf{inv}(R_i)\}$ for each $i \in [0..n]$. Note that, because R_0 and R_n are composite roles, so are P_0 and P_n. Now, if $P_0 = P_n$, the path $P_0, \ldots, P_n$ describes a cycle in $G_{\mathcal{R}}$ that contains composite roles. Otherwise, because $R_0 = R_n$ and $P_0, P_n \in \{R_0, \mathsf{inv}(R_0)\}$, we must have that $P_0 = R_0$ and $P_n = \mathsf{inv}(R_0)$, or vice versa. In either case, since $\langle P_0, P_1 \rangle$ is an edge in $G_{\mathcal{R}}$, we have that $\langle \mathsf{inv}(P_0), P_1 \rangle$ is also an edge in $G_{\mathcal{R}}$, and so the path $\mathsf{inv}(P_0), \ldots, P_n$ describes a cycle in $G_{\mathcal{R}}$ that contains composite roles, which is a contradiction.

Even though $G_{\mathcal{R}_1}$ does not contain cycles involving composite roles, the graph may

contain cycles that consist only of roles that are not composite. We next construct RBox $\mathcal{R}'$ by eliminating these cycles and we also compute a function rep that maps each role $R \in \mathsf{rol}_{\mathcal{R}_1}$ to a representative role $rep(R) \in \mathsf{rol}_{\mathcal{R}}$; we use function rep later to compute the knowledge $\mathcal{K}'$ and the query q'. To this end, let $G_{\mathcal{H}}$ be the directed graph that contains an edge $\langle S, R\rangle$ for each role inclusion $S \sqsubseteq R \in \mathcal{R}_1$; furthermore, let SCC be the set containing each maximal, strongly connected component Π of the graph $G_{\mathcal{H}}$. It is well-known that the set SCC can be computed in polynomial time. Please note that, because $\top_r \not\sqsubseteq^*_{\mathcal{R}} \bot_r$, for each $\Pi \in SCC$, we have that $\{\top_r, \bot_r\} \not\subseteq \Pi$. Next, for each role $R \in \mathsf{rol}_{\mathcal{R}_1}$, let $rep(R)$ be an arbitrary, but fixed representative of the strongly connected component $\Pi \in SCC$ containing R such that $rep(R) = \top_r$ if $\top_r \in \Pi$, and $rep(R) = \bot_r$ if $\bot_r \in \Pi$. Then, let $\mathcal{R}'$ be the following RBox.

$$\begin{aligned}\mathcal{R}' = &\{rep(R) \sqsubseteq R \mid R \in \mathsf{rol}_{\mathcal{R}_1}\} \\ &\cup \{rep(R_1) \cdots rep(R_n) \sqsubseteq rep(R) \mid R_1 \cdots R_n \sqsubseteq R \in \mathcal{R}\}\end{aligned}$$

It should be clear that the graph $\mathcal{G}_{\mathcal{R}'}$ associated with $\mathcal{R}'$ is acyclic. Furthermore, by the definition of $\mathcal{G}_{\mathcal{R}'}$, for each edge $\langle S, R\rangle$ in $G_{\mathcal{R}'}$ we have that $\langle \mathsf{inv}(S), R\rangle$ occurs in $G_{\mathcal{R}'}$ as well. Therefore, we can read off $\mathcal{G}_{\mathcal{R}'}$ the regular order that shows that $\mathcal{R}'$ is regular.

Finally, let $\mathcal{K}' = \mathcal{T}' \cup \mathcal{R}' \cup \mathcal{A}'$ where $\mathcal{T}'$ and $\mathcal{A}'$ are obtained from $\mathcal{T}$ and $\mathcal{A}$ by replacing each role $R \in \mathsf{rol}_R$ with its representative $rep(R)$. Similarly, let q' be the result of replacing each binary atom $R(s,t)$ with $rep(R)(s,t)$. Then, by the definition of $\mathcal{K}'$, each model $\mathcal{I}$ of $\mathcal{K}_1$ is a model of $\mathcal{K}'$. Similarly, each model $\mathcal{J}$ of $\mathcal{R}'$ can be expanded to a model $\mathcal{I}$ of $\mathcal{K}_1$ by interpreting each role $R \in \mathsf{rol}_{\mathcal{R}_1}$ as $R^{\mathcal{I}} = rl(R)^{\mathcal{J}}$. Therefore, we have that $\mathcal{K}_1 \models_{\mathcal{DL}} q$ if and only if $\mathcal{K}' \models_{\mathcal{DL}} \pi(q')$. □

Chapter 13

Outlook

In this thesis, we presented several novel complexity results and practicable algorithms for answering expressive queries over knowledge bases formulated in the $\mathcal{EL}$ family of DLs, the family of languages that logically underpin the EL profile of OWL 2.

Many of our algorithms and results are based on our translation of knowledge bases into datalog programs. This translation can be applied to any fragment of OWL 2 EL; and the translation preserves both knowledge base consistency and answers to instance queries. In contrast, evaluating CQs over the datalog program may produce answers that are unsound w.r.t. the knowledge base.

Based on our datalog translation, we then proposed a procedure for answering CQs over $\mathcal{ELHO}_\bot^+$ knowledge bases which runs in nondeterministic polynomial time in the input size, but runs in polynomial time in the size of the knowledge base. Moreover, by taking advantage of an innovative, succinct encoding of regular role inclusions using bounded-stack PDA, we extended the CQ answering algorithm for $\mathcal{ELHO}_\bot^+$ KBs to obtain the first PSPACE procedure for answering CQs over $\mathcal{ELRO}_\bot^+$ KBs with regular role inclusions. Thus, we closed two long-standing open questions on the complexity of CQ answering in $\mathcal{EL}$ with transitive roles and in OWL 2 EL.

To experimentally validate the practicability of our method for answering CQs, we have implemented a prototypical system, called EOLO, that implements the CQ answering algorithm for $\mathcal{ELHO}_\bot^+$. Our preliminary evaluation suggests that the overhead caused by materialising the consequences of the datalog program $\mathsf{D}_\mathcal{K}$ is manageable in many practical

situations. Furthermore, although some queries may be challenging, the majority of queries that we tested retrieve a moderate percentage of unsound candidate answers, and on average each candidate answer can be filtered in as few as a couple of microseconds for CQs that do not use transitive roles, and at most several milliseconds for CQs with transitive roles. Thus, we believe that our approach can provide an adequate practical basis for CQ answering over OWL 2 EL KBs.

As answering CQs over KBs formulated in the $\mathcal{EL}$ family of DLs is a computationally expensive task, we also studied the complexity of answering acyclic CQs. Our results suggest that $\mathcal{ELH}_\bot$ is essentially the largest member of the $\mathcal{EL}$ family of DLs for which answering acyclic CQs is tractable. Furthermore, we show that CQ answering over $\mathcal{ELRO}^+_\bot$ knowledge bases is PSpace-hard even when just the role inclusions are considered as part of the input (i.e., the query and all other parts of the knowledge base are fixed); thus, we show that the problem is PSpace-hard also in KB complexity.

We also studied the problem of answering navigational queries and showed that positive, converse-free GXQs and CGXQs can be answered over OWL 2 EL KBs in PTime and PSpace, respectively; this is interesting because Bienvenu et al. [11] proved that answering positive GXQs over $\mathcal{ELH}_\bot$ KBs is ExpTime-complete; hence, adding the converse operator increases the complexity of graph XPath queries. Our results thus suggest that, at least from a theoretical perspective, positive, converse-free GXQs and CGXQs are appealing as query languages for OWL 2 EL knowledge bases.

Finally, we reviewed the existing regularity restriction on role inclusions in the OWL 2 specification [78], and showed that this condition does not ensure that each language induced by a knowledge base is regular. To solve this issue, we proposed a revised syntactic condition that correctly ensures the regularity of the induced languages; furthermore, in the same way as the original condition by Motik et al. [78], our revised restriction generalises the well-known regularity condition by Horrocks and Sattler [51], extends to role hierarchies, and allows for the specification of the compositional properties of inverse roles.

Table 13.1: The complexity landscape for conjunctive and graph queries over OWL 2 EL

(a) The complexity landscape of acyclic CQ answering (all are completeness results)

	$\mathcal{ELH}_\bot$	$\mathcal{ELHO}_\bot$	$\mathcal{ELHO}_\bot^+$	$\mathcal{ELRO}_\bot^+$
arborescent	PTime [10]	PTime (Th. 9.8)	NP (Th. 6.14)	PSpace (Th. 9.8)
acyclic	PTime [10]	NP (Th. 9.8)	NP (Th. 6.14)	PSpace [66]

(b) The complexity landscape of CQ answering (all are completeness results)

	$\mathcal{ELH}_\bot$	$\mathcal{ELHO}_\bot^+$	$\mathcal{ELRO}_\bot^+$	Horn-$\mathcal{SHOIQ}$	Horn-$\mathcal{SROIQ}$
data	PTime [86]	PTime (Th. 6.14)	PTime (Th. 9.8)	PTime [83]	PTime [83]
combined	NP [86]	NP (Th. 6.14)	PSpace (Th. 9.8)	ExpTime [83]	2ExpTime [83]

(c) The complexity of answering graph XPath queries over $\mathcal{ELRO}_\bot^+$ knowledge bases ('c' means 'complete', and 'h' means 'hard')

	positive converse-free GXQs	positive converse-free CGXQs	positive GXQs	path-positive GXQs	GXQs
data	PTime-c (Th. 11.2)	PTime-c (Th. 11.2)	PTime-h [11]	coNP-h [61]	undec. [61]
combined	PTime-c (Th. 11.2)	PSpace-c (Th. 11.2)	ExpTime-h [11]	ExpTime-h [11]	undec. [61]

13.1 The Complexity of Query Answering in OWL 2 EL

We next harness the novel results presented in this thesis and review the complexity landscape of answering acyclic and unrestricted CQs, as well as navigational queries in various DLs related to OWL 2 EL.

Table 13.1a summarises the combined complexity of answering arborescent and acyclic CQs in important members of the $\mathcal{EL}$ family of DLs. The tractability result for acyclic CQ answering in $\mathcal{ELH}_\bot$ is due to Bienvenu et al. [10], while the PSPACE-hardness for acyclic CQ answering in $\mathcal{ELRO}^+_\bot$ is due to Krötzsch et al. [66]; all other results have been proved in this thesis. As one can see, extending $\mathcal{ELH}_\bot$ with nominals increases the complexity of acyclic CQ answering, even though answering arborescent queries remains tractable. In Theorem 10.6, we showed that extending $\mathcal{EL}$ with transitive or reflexive roles immediately leads to the intractability of arborescent (and thus acyclic) CQ answering; furthermore, extending $\mathcal{EL}$ with regular role makes the problem PSPACE-hard and, by our PSPACE lower bound, this increase is solely due to role inclusions.

Table 13.1b summarises the complexity landscape of answering unrestricted CQs in various DLs related to $\mathcal{ELRO}^+_\bot$ (and thus to OWL 2 EL). Here, Horn-$\mathcal{SHOIQ}$ extends $\mathcal{ELHO}^+_\bot$ with inverse roles and Horn qualified number restrictions, and Horn-$\mathcal{SROIQ}$ extends Horn-$\mathcal{SHOIQ}$ with regular role inclusions. The results for these logics are due to Ortiz et al. [83], whereas the results for $\mathcal{ELH}_\bot$ are due to Rosati [86]. CQ answering is PTIME-complete in data complexity in all cases, which is essentially due to the fact that all of these logics are Horn so no disjunctive reasoning is needed. For the combined complexity, the table illustrates how the presence of different constructs affects the complexity of answering CQs. In particular, we showed that extending $\mathcal{ELH}_\bot$ with transitive roles, reflexive roles, and nominals does not increase the complexity of CQ answering. This is interesting because role transitivity suffices to express simple graph properties such as reachability, and it is a known source of complexity of CQ answering [28, 39]. In contrast, extending $\mathcal{ELHO}^+_\bot$ with regular role inclusions increases the complexity from NP to PSPACE. Furthermore, extending $\mathcal{ELHO}^+_\bot$ with inverse roles increases the complexity from NP to EXPTIME. Finally, extending $\mathcal{ELRO}^+_\bot$ with inverse roles increases the complexity from PSPACE to 2EXPTIME.

Finally, Table 13.1c summarises the complexity landscape of answering graph XPath queries over $\mathcal{ELRO}^{+}_{\bot}$ KBs. As one can see, adding the converse operator increases the combined complexity of GXQs to ExpTime [11]. Moreover, adding negation over node tests increases the data complexity of GXQs to coNP, whereas adding negation over path expressions leads to the undecidability even when the TBox and the RBox are empty and the query is considered to be fixed [61]. In contrast, existential quantification over paths does not increase the complexity: answering positive, converse-free (C)GXQs over $\mathcal{ELRO}^{+}_{\bot}$ knowledge bases is as difficult as answering (C)RPQs over $\mathcal{EL}$ knowledge bases [9].

13.2 Future Work

We anticipate several directions for our future work. An important theoretical and practical challenge is to develop novel optimisations that reduce the number of nondeterministic choices required by the CQ answering algorithm for OWL 2 EL, and to implement the resulting algorithm in order to evaluate its practicability.

Another theoretical challenge is to investigate the use of top-down query evaluation techniques—such as magic sets [1] or SLG resolution [19]. Moreover, tighter integration of the detection of unsound answers with the query evaluation algorithms should make it possible to eagerly detect unsound answers (i.e., before the query is fully evaluated); thus reducing the overhead caused by queries that generate a large number of unsound answers.

As our datalog translation of knowledge bases can be used to check the consistency of, and answer database-like queries over OWL 2 EL knowledge bases, an important open problem is whether our datalog encoding can also be used to efficiently check concept subsumptions, thus leading the way towards a unified reasoning system for OWL 2 EL.

Finally, as static query analysis is a fundamental task in query optimisation, we shall study the containment problem for graph queries under OWL 2 EL constraints.

Bibliography

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. ISBN 0-201-53771-0.

[2] A. V. Aho, C. Beeri, and J. D. Ullman. The theory of joins in relational databases. *ACM Trans. Database Syst.*, 4(3):297–314, September 1979. ISSN 0362-5915. doi: 10.1145/320083.320091. URL http://doi.acm.org/10.1145/320083.320091.

[3] Marcella Anselmo, Dora Giammarresi, and Stefano Varricchio. Finite automata and non-self-embedding grammars. In *Proceedings of the 7th International Conference on Implementation and Application of Automata*, CIAA'02, pages 47–56. Springer-Verlag, 2003. ISBN 3-540-40391-4. URL http://dl.acm.org/citation.cfm?id=1756384.1756390.

[4] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*, pages 364–369, Edinburgh, UK, July 30–August 5 2005. Morgan Kaufmann Publishers.

[5] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope further. In *In Proceedings of the OWLED 2008 DC Workshop on OWL: Experiences and Directions*, 2008.

[6] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2010. Paperback edition.

[7] Pablo Barceló. Querying graph databases. In *Proceedings of the 32nd ACM SIGMOD-*

SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, pages 175–188. ACM, 2013.

[8] Chris Barrett, Riko Jacob, and Madhav Marathe. Formal-language-constrained path problems. *SIAM Journal on Computing*, 30(3):809–837, May 2000. ISSN 0097-5397. doi: 10.1137/S0097539798337716. URL http://dx.doi.org/10.1137/S0097539798337716.

[9] Meghyn Bienvenu, Magdalena Ortiz, and Mantas Simkus. Conjunctive regular path queries in lightweight description logics. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. IJCAI/AAAI, 2013.

[10] Meghyn Bienvenu, Magdalena Ortiz, Mantas Simkus, and Xiao Guohui. Tractable queries for lightweight description logics. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. IJCAI/AAAI, 2013.

[11] Meghyn Bienvenu, Diego Calvanese, Magdalena Ortiz, and Mantas Simkus. Nested regular path queries in description logics. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014*. AAAI Press, 2014.

[12] Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *Journal of Artificial Intelligence Research*, 48:115–174, 2013. ISSN 10769757. doi: 10.1613/jair.3873.

[13] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. In *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, KR 2006*, pages 260–270. AAAI Press, 2006.

[14] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

[15] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Regular path queries in expressive description logics with nominals. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 714–720, 2009.

[16] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. The MASTRO system for ontology-based data access. *Semantic Web*, 2(1):43–53, 2011.

[17] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Answering regular path queries in expressive description logics via alternating tree-automata. *Inf. Comput.*, 237:12–55, 2014. doi: 10.1016/j.ic.2014.04.002. URL http://dx.doi.org/10.1016/j.ic.2014.04.002.

[18] A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In J. E. Hopcroft, E. P. Friedman, and M. A. Harrison, editors, *Proc. of the 9th annual ACM Symposium on Theory of Computing (STOC '77)*, pages 77–90, Boulder, CO, USA, May 2–4 1977. ACM Press.

[19] Weidong Chen and David S. Warren. Query evaluation under the well-founded semantics. In *Proceedings of the Twelfth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '93, pages 168–179, New York, NY, USA, 1993. ACM. ISBN 0-89791-593-3. doi: 10.1145/153850.153865. URL http://doi.acm.org/10.1145/153850.153865.

[20] Alexandros Chortaras, Despoina Trivela, and Giorgos Stamou. Goal-oriented query rewriting for OWL 2 QL. In Riccardo Rosati, Sebastian Rudolph, and Michael Zakharyaschev, editors, *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*, volume 745. CEUR-WS.org, 2011.

[21] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms (3. ed.)*. MIT Press, 2009. ISBN 978-0-262-03384-8. URL http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=11866.

[22] Isabel F. Cruz, Alberto O. Mendelzon, and Peter T. Wood. A graphical query language supporting recursion. In *Proceedings of the 1987 ACM SIGMOD International Conference on Management of Data*, SIGMOD '87, pages 323–330, New York, NY, USA, 1987. ACM. ISBN 0-89791-236-5. doi: 10.1145/38713.38749. URL http://doi.acm.org/10.1145/38713.38749.

[23] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter F. Patel-Schneider, and Ulrike Sattler. OWL 2: The next step for OWL. *J. Web Sem.*, 6(4): 309–322, 2008.

[24] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. *ACM Comput. Surv.*, 33(3):374–425, 2001. ISSN 03600300. doi: 10.1145/502807.502810. URL http://portal.acm.org/citation.cfm?doid=502807.502810.

[25] Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. MASTRO: A reasoner for effective ontology-based data access. In *Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE-2012)*. CEUR-WS.org, 2012.

[26] Steven DeRose and James Clark. XML path language (XPath) version 1.0. W3C recommendation, W3C, November 1999. http://www.w3.org/TR/1999/REC-xpath-19991116.

[27] Sebastian Derriere, André Richard, and Andrea Preite-Martinez. An ontology of astronomical object types for the virtual observatory. *Proceedings of the International Astronomical Union*, 2(14), 2006. ISSN 1743-9213. doi: 10.1017/S174392130701201X.

[28] Thomas Eiter, Carsten Lutz, Magdalena Ortiz, and Mantas Simkus. Query answering in description logics with transitive roles. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 759–764, 2009.

[29] Thomas Eiter, Magdalena Ortiz, and Mantas Simkus. Conjunctive query answering in the description logic $\mathcal{SH}$ using knots. *J. Comput. Syst. Sci.*, 78(1):47–85, 2012.

[30] Thomas Eiter, Magdalena Ortiz, Mantas Simkus, Trung-Kien Tran, and Guohui Xiao. Query rewriting for Horn-$\mathcal{SHIQ}$ plus rules. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. AAAI Press, 2012.

[31] Ronald Fagin, Phokion G. Kolaitis, and Lucian Popa. Data exchange: Getting to the core. *ACM Trans. Database Syst.*, 30(1):174–210, March 2005. ISSN 0362-5915. doi: 10.1145/1061318.1061323. URL http://doi.acm.org/10.1145/1061318.1061323.

[32] Wenfei Fan. Graph pattern matching revised for social network analysis. In *15th International Conference on Database Theory, ICDT '12,*, pages 8–21. ACM, 2012.

[33] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. ISBN 0-7167-1044-7.

[34] Viliam Geffert, Carlo Mereghetti, and Beatrice Palano. More concise representation of regular languages by automata and regular expressions. *Information and computation*, 208(4):385–394, 2010.

[35] Martin Giese, Diego Calvanese, Peter Haase, Ian Horrocks, Yannis Ioannidis, Heralk Kllapi, Manolis Koubarakis, Maurizio Lenzerini, Ralf Möller, Mariano Rodriguez-Muro, Özgür Özcep, Riccardo Rosati, Rudolf Schlatte, Michael Schmidt, Ahmet Soylu, and Arild Waaler. Scalable end-user access to big data. In Rajendra Akerkar, editor, *Big Data Computing*. CRC Press, 2013.

[36] Birte Glimm, Carsten Lutz, Ian Horrocks, and Ulrike Sattler. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. *J. Artif. Intell. Res. (JAIR)*, 31:157–204, 2008.

[37] Georg Gottlob and Thomas Schwentick. Rewriting ontological queries into small nonrecursive datalog programs. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012*. AAAI Press, 2012.

[38] Georg Gottlob, Nicola Leone, and Francesco Scarcello. The complexity of acyclic conjunctive queries. *Journal of the ACM*, 48(3):431–498, May 2001. ISSN 00045411.

doi: 10.1145/382780.382783. URL http://portal.acm.org/citation.cfm?doid=382780.382783.

[39] Georg Gottlob, Andreas Pieris, and Lidia Tendera. Querying the guarded fregment with transitivity. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming–40th International Colloquium, ICALP 2013*, volume 7966, pages 287–298. Springer, 2013.

[40] Georg Gottlob, Stanislav Kikot, Roman Kontchakov, Vladimir Podolskii, Thomas Schwentick, and Michael Zakharyaschev. The price of query rewriting in ontology-based data access. *Artificial Intelligence*, 213:42–59, aug 2014. ISSN 00043702. doi: 10.1016/j.artint.2014.04.004. URL http://linkinghub.elsevier.com/retrieve/pii/S0004370214000459.

[41] Georg Gottlob, Marco Manna, and Andreas Pieris. Polynomial combined rewritings for existential rules. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014*. AAAI Press, 2014.

[42] Jennifer Green, Catherine Dolbear, Glen Hart, John Goodwin, and Paula Engelbrecht. Creating a semantic integration system using spatial data. In *7th International Semantic Web Conference, ISWC 2008*, 2008.

[43] Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker. Description logic programs: Combining logic programs with description logic. In *Proceedings of the 12th international conference on World Wide Web*, pages 48–57, 2003. ISBN 1581136803. URL http://dl.acm.org/citation.cfm?id=775160.

[44] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. LUBM: A benchmark for OWL knowledge base systems. *Web Semantics*, 3(2-3):158–182, 2005. ISSN 15708268. doi: 10.1016/j.websem.2005.06.005.

[45] Claudio Gutierrez, Carlos A. Hurtado, Alberto O. Mendelzon, and Jorge Pérez. Foundations of Semantic Web databases. *J. Comput. Syst. Sci.*, 77(3):520–541, 2011.

[46] Victor Gutiérrez-Basulto, Yazmin Ibanez-Garcia, Roman Kontchakov, and Egor V Kostylev. Queries with negation and inequality over lightweight ontologies. *Journal of Web Semantics*, 2014. URL http://www.informatik.uni-bremen.de/tdki/research/papers/2014/GuIbKoKo14.pdf.

[47] David Harel. Dynamic Logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic Vol. II*, pages 497–604. Reidel Publishing Company, 1984.

[48] David Harel, Jerzy Tiuryn, and Dexter Kozen. *Dynamic Logic*. MIT Press, Cambridge, MA, USA, 2000. ISBN 0262082896.

[49] Tony Hey and Anne E Trefethen. Cyberinfrastructure for e-Science. *Science (New York, N.Y.)*, 308(5723):817–821, 2005. ISSN 0036-8075. doi: 10.1126/science.1110410.

[50] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation - international edition (2. ed)*. Addison-Wesley, 2003. ISBN 978-0-321-21029-6.

[51] Ian Horrocks and Ulrike Sattler. Decidability of $\mathcal{SHIQ}$ with complex role inclusion axioms. *Artificial Intelligence*, 160(1–2):79–104, December 2004. ISSN 09401121.

[52] Ian Horrocks and Sergio Tessaris. Querying the Semantic Web: a formal approach. In *Semantic Web-ISWC 2002*, pages 177–191, 2002. ISBN 3-540-43760-6. doi: 10.1007/3-540-48005-6_15. URL http://www.springerlink.com/index/XAXBYMXFK0KX5LXG.pdf.

[53] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The even more irresistible $\mathcal{SROIQ}$. In *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 57–67. AAAI Press, 2006.

[54] Yevgeny Kazakov. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008*, pages 274–284. AAAI Press, 2008.

[55] Stanislav Kikot, Roman Kontchakov, and Michael Zakharyaschev. Conjunctive query answering with OWL 2 QL. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012*, 2012.

[56] SC Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, 1956.

[57] Matthias Klusch, Benedikt Fries, and Katia Sycara. Automated semantic web service discovery with OWLS-MX. In *AAMAS 06 Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 915–922, 2006. ISBN 1595933034. doi: 10.1145/1160633.1160796. URL http://portal.acm.org/citation.cfm?id=1160796.

[58] Mélanie König, Michel Leclere, Marie-Laure Mugnier, and Mickael Thomazo. A sound and complete backward chaining algorithm for existential rules. In *RR 2012*, pages 122–138, 2012. URL http://link.springer.com/chapter/10.1007/978-3-642-33203-6_10.

[59] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to query answering in DL-Lite. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010*, 2010. URL http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1282.

[60] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to ontology-based data access. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2656–2661. IJCAI/AAAI, 2011.

[61] Egor V. Kostylev, Juan L. Reutter, and Domagoj Vrgoc. XPath for DL ontologies. In *Proc. of the 29th AAAI Conference on Artificial Intelligence (AAAI-15)*, 2015.

[62] Dexter Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science, FOCS 1977*, pages 254–266, 1977.

[63] Adila Krisnadhi and Carsten Lutz. Data complexity in the EL family of description logics. In *Proceedings of the 14th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning*, 2007.

[64] Markus Krötzsch. Efficient rule-based inferencing for OWL EL. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI 2011*. AAAI Press/IJCAI, 2011. 2668–2673.

[65] Markus Krötzsch and Sebastian Rudolph. Conjunctive queries for EL with composition of roles. In *Proceedings of the 2007 International Workshop on Description Logics (DL2007), Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June, 2007*, volume 250 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007. URL http://ceur-ws.org/Vol-250/paper_58.pdf.

[66] Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. Conjunctive queries for a tractable fragment of OWL 1.1. In *Proceedings of the 6th International Semantic Web Conference (ISWC'07)*, volume 4825 of *LNCS*, pages 310–323. Springer, 2007.

[67] Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. ELP: Tractable rules for OWL 2. In *Proceedings of the 7th International Semantic Web Conference, ISWC 2008*, pages 649–664, 2008.

[68] Hector J. Levesque and Ronald J. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3:78–93, 1987. doi: 10.1111/j.1467-8640.1987.tb00176.x. URL http://dx.doi.org/10.1111/j.1467-8640.1987.tb00176.x.

[69] Anita C. Liang, Boris Lauser, Margherita Sini, Johannes Keizer, and Stephen Katz. From AGROVOC to the agricultural ontology service / concept server An OWL model for managing ontologies in the agricultural domain. In *Dublin Core Conference Proceedings*, volume 216, 2006. ISBN 3906570568.

[70] Leonid Libkin, Wim Martens, and Domagoj Vrgoč. Querying graph databases with

XPath. In *Joint 2013 EDBT/ICDT Conferences, ICDT '13 Proceedings*, pages 129–140. ACM, 2013.

[71] Carsten Lutz. The complexity of conjunctive query answering in expressive description logics. In *Automated Reasoning*, 2008. URL http://link.springer.com/chapter/10.1007/978-3-540-71070-7_16.

[72] Carsten Lutz, David Toman, and Frank Wolter. Conjunctive query answering in the description logic EL using a relational database system. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 2070–2075, 2009.

[73] Carsten Lutz, Inanç Seylan, David Toman, and Frank Wolter. The combined approach to OBDA: taming role hierarchies using filters. In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference*, volume 8218 of *Lecture Notes in Computer Science*, pages 314–330. Springer, 2013. doi: 10.1007/978-3-642-41335-3_20. URL http://dx.doi.org/10.1007/978-3-642-41335-3_20.

[74] David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv. Testing implications of data dependencies. *ACM Trans. Database Syst.*, 4(4):455–469, 1979. ISSN 03625915. doi: 10.1145/320107.320115.

[75] Bruno Marnette. Generalized schema-mappings: from termination to tractability. In *Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009*, pages 13–22. ACM, 2009.

[76] Marvin Minsky. A framework for representing knowledge. In P.H. Winston, editor, *The Psychology of Computer Vision*, pages 211–278. McGraw-Hill, NY, 1975. URL http://18.7.29.232/handle/1721.1/6089.

[77] José Mora, Riccardo Rosati, and Óscar Corcho. kyrie2: Query rewriting under extensional constraints in *ELHIO*. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference*, pages 568–583, 2014. doi: 10.1007/978-3-319-11964-9_36. URL http://dx.doi.org/10.1007/978-3-319-11964-9_36.

[78] Boris Motik, Peter Patel-Schneider, and Bijan Parsia. OWL 2 web ontology language structural specification and functional-style syntax (second edition). W3C recommendation, W3C, December 2012. http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/.

[79] Boris Motik, Yavor Nenov, Robert Piro, Ian Horrocks, and Dan Olteanu. Parallel materialisation of datalog programs in centralised, main-memory RDF systems. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, 2014.

[80] Boris Motik, Yavor Nenov, Robert Piro, and Ian Horrocks. Incremental update of datalog materialisation: The backward/forward algorithm. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1560–1568. AAAI Press, 2015.

[81] Adrian Onet. The chase procedure and its applications in data exchange. In *Data Exchange, Integration, and Streams*, volume 5 of *Dagstuhl Follow-Ups*, pages 1–37. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013. ISBN 978-3-939897-61-3. doi: http://dx.doi.org/10.4230/DFU.Vol5.10452.1. URL http://drops.dagstuhl.de/opus/volltexte/2013/4288.

[82] Magdalena Ortiz, Diego Calvanese, and Thomas Eiter. Data complexity of query answering in expressive description logics via tableaux. *J. Autom. Reasoning*, 41(1): 61–98, 2008.

[83] Magdalena Ortiz, Sebastian Rudolph, and Mantas Simkus. Query answering in the Horn fragments of the description logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 1039–1044, 2011.

[84] Christos H Papadimitriou. *Computational Complexity*, volume 11. Addison Wesley, 1994. ISBN 0201530821. doi: 10.1006/jcom.1995.1011. URL http://portal.acm.org/citation.cfm?id=1074233.

[85] Héctor Pérez-Urbina, Boris Motik, and Ian Horrocks. Tractable query answering and rewriting under description logic constraints. *J. Applied Logic*, 8(2):186–209, 2010.

[86] Riccardo Rosati. On conjunctive query answering in $\mathcal{EL}$. In *Proceedings of the 2007 International Workshop on Description Logics (DL2007)*, 2007.

[87] Riccardo Rosati. The limits of querying ontologies. In *Database Theory - ICDT 2007, 11th International Conference*, pages 164–178, 2007.

[88] Riccardo Rosati. Prexto: Query rewriting under extensional constraints in DL-Lite. In *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012*, volume 7295 LNCS, pages 360–374. Springer, 2012. ISBN 9783642302831. doi: 10.1007/978-3-642-30284-8_31.

[89] M. Ross Quillan. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, 12(5):410–430, 1967.

[90] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. Syst. Sci.*, 4:177–192, 1970. ISSN 00220000. doi: 10.1016/S0022-0000(70)80006-X.

[91] Stefan Schulz, Ronald Cornet, and Kent Spackman. Consolidating SNOMED CT's ontological commitment. *Applied Ontology*, 6:1–11, 2011. doi: 10.3233/AO-2011-0084. URL http://iospress.metapress.com/index/P614602542457GWP.pdf.

[92] František Simančík. Elimination of complex RIAs without automata. In *Proceedings of the 2012 International Workshop on Description Logics, DL-2012*, 2012.

[93] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *J. Web Sem.*, 5(2):51–53, 2007.

[94] Giorgio Stefanoni and Boris Motik. Conjunctive queries over $\mathcal{EL}$ with transitive and reflexive roles. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1611–1617. AAAI Press, 2015.

[95] Giorgio Stefanoni, Boris Motik, and Ian Horrocks. Introducing nominals to the combined query answering approaches for $\mathcal{EL}$. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[96] Giorgio Stefanoni, Boris Motik, Markus Krötzsch, and Sebastian Rudolph. The complexity of answering conjunctive and navigational queries over OWL 2 EL knowledge bases. *J. Artif. Intell. Res. (JAIR)*, 51:645–705, 2014.

[97] Giorgos Stoilos. Hydrowl: A hybrid query answering system for OWL 2 DL ontologies. In *Proc. 8th Int. Conf. Web Reason. Rule Syst. (RR 2014)*, 2014.

[98] Dmitry Tsarkov and Ian Horrocks. Fact++ description logic reasoner: System description. In *Automated Reasoning, Third International Joint Conference, IJCAR 2006*, pages 292–297, 2006.

[99] Jacopo Urbani, Frank van Harmelen, Stefan Schlobach, and Henri E. Bal. QueryPIE: Backward reasoning for OWL Horst over very large knowledge bases. In *The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference*, pages 730–745, 2011.

[100] Moshe Y. Vardi. The complexity of relational query languages (extended abstract). In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, STOC '82, pages 137–146, New York, NY, USA, 1982. ACM. ISBN 0-89791-070-2. doi: 10.1145/800070.802186. URL http://doi.acm.org/10.1145/800070.802186.

[101] Tassos Venetis, Giorgos Stoilos, and Giorgos B. Stamou. Incremental query rewriting for OWL 2 QL. In *Proceedings of the 2012 International Workshop on Description Logics*, 2012.

[102] Tassos Venetis, Giorgos Stoilos, and Giorgos Stamou. Query rewriting under query extensions for OWL 2 QL ontologies. *J. Data Semant.*, 3:1–23, 2014. ISSN 16130073. doi: 10.1007/s13740-012-0017-6.

[103] Roberto De Virgilio, Giorgio Orsi, Letizia Tanca, and Riccardo Torlone. NYAYA: A system supporting the uniform management of large sets of semantic data. In *IEEE*

28th International Conference on Data Engineering (ICDE 2012), pages 1309–1312, 2012. doi: 10.1109/ICDE.2012.133. URL http://dx.doi.org/10.1109/ICDE.2012.133.

[104] Artem Vorobiev and Jun Han. Security attack ontology for Web services. In *2006 2nd International Conference on Semantics Knowledge and Grid, SKG*, 2006. ISBN 0769532055. doi: 10.1109/SKG.2006.85.

[105] Michael Wessel. Obstacles on the way to qualitative spatial reasoning with description logics: Some undecidability results. In *Working Notes of the 2001 International Description Logics Workshop (DL-2001)*, volume 49. CEUR-WS.org, 2001.

[106] Mihalis Yannakakis. Algorithms for acyclic database schemes. In *Proceedings of the seventh international conference on Very Large Data Bases*, volume 7, pages 82–94, 1981.

[107] Yujiao Zhou, Yavor Nenov, Bernardo Cuenca Grau, and Ian Horrocks. Complete query answering over Horn ontologies using a triple store. In *International Semantic Web Conference 2013*, volume 8218 LNCS, pages 720–736, 2013. ISBN 9783642413346. doi: 10.1007/978-3-642-41335-3_45.

[108] Yujiao Zhou, Yavor Nenov, Bernardo Cuenca Grau, and Ian Horrocks. Pay-as-you-go OWL query answering using a triple store. In *Proc. 28th Conf. Artif. Intell. (AAAI 14)*, pages 1142–1148. AAAI Press, 2014. ISBN 9781577356783.