Answering Regular Path Queries over $\mathcal{SQ}$ Ontologies

Víctor Gutiérrez-Basulto
Cardiff University, UK
gutierrezbasultov@cardiff.ac.uk

Yazmín Ibáñez-García
TU Wien, Austria
yazmin.garcia@tuwien.ac.at

Jean Christoph Jung
Universität Bremen, Germany
jeanjung@uni-bremen.de

Abstract

We study query answering in the description logic $\mathcal{SQ}$ supporting qualified number restrictions on both transitive and non-transitive roles. Our main contributions are a tree-like model property for $\mathcal{SQ}$ knowledge bases and, building upon this, an optimal automata-based algorithm for answering positive existential regular path queries in 2EXPTIME.

1 Introduction

The use of ontologies to access data has gained a lot of popularity in various research fields such as knowledge representation and reasoning, and databases. In the ontology-based data access (OBDA) scenario, ontologies are often encoded using description logic languages (DLs); as a consequence, a large amount of research on the query answering problem (QA) over DL ontologies has been conducted. In particular, several efforts have been put into the study of the query answering problem in DLs featuring *transitive roles* and *number restrictions* (Glimm, Horrocks, and Sattler 2008; Glimm et al. 2008; Eiter et al. 2009; Calvanese, Eiter, and Ortiz 2009; 2014). However, in all these works the application of number restrictions to transitive roles is forbidden. This is also reflected in the fact that the W3C ontology language OWL 2 does not allow for this interaction.[1] Unfortunately, this comes as a shortcoming in crucial DL application areas like medicine and biology in which many terms are defined and classified according to the number of components they contain or have as a part, in a transitive sense (Wolstencroft et al. 2005; Rector and Rogers 2006; Stevens et al. 2007). For instance, the ontology $\mathcal{T}$ below describes that the human heart has as a part (hPt) exactly one mitral valve (MV), a left atrium (LA) and a left ventricle (LV); and the latter two (enforced to be distinct) also have as a part a mitral valve. Thus, the left atrium and left ventricle have to share the mitral valve.

$$\mathcal{T} = \{\ \mathsf{Heart} \sqsubseteq (= 1\ \mathsf{hPt.MV}) \sqcap \exists \mathsf{hPt.LA} \sqcap \exists \mathsf{hPt.LV}, \\ \mathsf{LV} \sqcap \mathsf{LA} \sqsubseteq \bot, \quad \mathsf{LV} \sqsubseteq \exists \mathsf{hPt.MV}, \quad \mathsf{LA} \sqsubseteq \exists \mathsf{hPt.MV}\ \}.$$

The lack of investigations of query answering in DLs of this kind is partly because (i) the interaction of these features with other traditional constructors often leads to undecidability of the standard reasoning tasks (e.g., satisfiability) (Horrocks, Sattler, and Tobies 2000); and (ii) for those DLs known to be decidable, such as $\mathcal{SQ}$ and $\mathcal{SOQ}$ (Kazakov, Sattler, and Zolin 2007; Kaminski and Smolka 2010), only recently tight complexity bounds were obtained (Gutiérrez-Basulto, Ibáñez-García, and Jung 2017a). Moreover, these features, even with restricted interaction, pose additional challenges for devising decision procedures since they lead to the loss of properties, such as the tree model property, which make the design of algorithms for QA simpler. Clearly, these issues are exacerbated if number restrictions are imposed on transitive roles.

Traditionally, most of the research in OBDA has focused on answering conjunctive queries. However, *navigational* queries have recently gained a lot of attention (Stefanoni et al. 2014; Bienvenu, Ortiz, and Simkus 2015; Baget et al. 2017) since they are key in various applications. For instance, in biomedicine they are used to retrieve specific paths from protein, cellular and disease networks (Dogrusoz et al. 2009; Lysenko et al. 2016). A prominent class of navigational queries is that of *regular path queries* (Florescu, Levy, and Suciu 1998), where paths are specified by a regular expression. Indeed, motivated by applications in the semantic web, the latest W3C standard SPARQL 1.1 includes property paths, related to regular expressions.

The objective of this paper is to start the research on query answering in DLs supporting qualified number restrictions over transitive roles. We study the entailment problem of *positive existential two-way regular path queries* (Calvanese et al. 2000) over $\mathcal{SQ}$ ontologies, thus generalizing both conjunctive and regular path queries. To this end, we pursue an *automata-based approach for query answering* using two-way alternating tree automata (2ATA) (Vardi 1998). This roughly consists of three steps (Calvanese, Eiter, and Ortiz 2014): (i) show that, if a query φ is not entailed by the knowledge base $\mathcal{K}$, there is a *tree-like* interpretation witnessing this, (ii) devise an automaton $\mathfrak{A}_\mathcal{K}$ which accepts precisely the tree-like interpretations of $\mathcal{K}$, (iii) devise an automaton $\mathfrak{A}_\varphi$ which accepts a tree-like interpretation iff it satisfies φ. Query entailment is then reduced to the question whether $\mathfrak{A}_\mathcal{K}$ accepts a tree that is not accepted by $\mathfrak{A}_\varphi$. In this paper, we significantly adapt and extend each step to $\mathcal{SQ}$, resulting in an algorithm running in 2EXPTIME, even

[1] https://www.w3.org/TR/webont-req/

for *binary* coding of numbers. A matching lower bound follows from positive existential QA in $\mathcal{ALC}$ (Calvanese, Eiter, and Ortiz 2014). More precisely, for step (i) we develop the notion of *canonical tree decompositions* which intuitively are tree decompositions tailored to handle the interaction of transitivity and number restrictions. We then show via a novel unraveling operation for $\mathcal{SQ}$ that, if the query is not entailed, there is a witness interpretation which has a canonical tree decomposition of width bounded exponentially in the size of $\mathcal{K}$, cf. Section 3. These canonical tree decompositions are crucial in order to construct a small 2ATA $\mathfrak{A}_\mathcal{K}$ in step (ii), which is done in Section 4.1. For step (iii), we propose in Section 4.2 a novel technique for answering regular path queries directly using a 2ATA $\mathfrak{A}_\varphi$ since a naive application of the techniques from (Calvanese, Eiter, and Ortiz 2014) does not lead to optimal complexity, because of the large width of the decompositions.

An extended version with appendix can be found under www.informatik.uni-bremen.de/tdki/research/papers.html.

2 Preliminaries

Syntax. We consider a vocabulary consisting of countably infinite disjoint sets of *concept names* $\mathsf{N_C}$, *role names* $\mathsf{N_R}$, and *individual names* $\mathsf{N_I}$, and assume that $\mathsf{N_R}$ is partitioned into two countably infinite sets of *non-transitive role names* $\mathsf{N_R^{nt}}$ and *transitive role names* $\mathsf{N_R^t}$. The syntax of $\mathcal{SQ}$-*concepts* C, D is given by the rule

$$C, D ::= A \mid \neg C \mid C \sqcap D \mid (\leqslant n\, r.C)$$

where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and n is a number given in binary. We use $(\geqslant n\, r.C)$ as an abbreviation for $\neg(\leqslant n{-}1\, r.C)$, and other standard abbreviations like $\bot$, $\top$, $C \sqcup D$, $\exists r.C$, $\forall r.C$. Concepts of the form $(\leqslant n\ r.C)$ and $(\geqslant n\ r.C)$ are called *at-most restrictions* and *at-least restrictions*, respectively.

An *$\mathcal{SQ}$-TBox (ontology) $\mathcal{T}$* is a finite set of *concept inclusions $C \sqsubseteq D$* where C, D are $\mathcal{SQ}$-concepts. An *ABox* is a finite set of *concept* and *role assertions* of the form $A(a)$, $r(a, b)$ where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$ and $\{a, b\} \subseteq \mathsf{N_I}$; $\mathsf{ind}(\mathcal{A})$ denotes the set of individual names occurring in $\mathcal{A}$. A *knowledge base (KB)* is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.

Semantics. An *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of a non-empty *domain* $\Delta^\mathcal{I}$ and an *interpretation function* $\cdot^\mathcal{I}$ mapping concept names to subsets of the domain and role names to binary relations over the domain such that transitive role names are mapped to transitive relations. The interpretation function is extended to complex concepts by defining $(\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I}$, $(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$, and

$$(\leqslant n\, r.C)^\mathcal{I} = \{d \in \Delta^\mathcal{I} \mid |\{e \in C^\mathcal{I} \mid (d, e) \in r^\mathcal{I}\}| \leq n\}.$$

For ABoxes $\mathcal{A}$ we adopt the *standard name assumption (SNA)*, that is, $a^\mathcal{I} = a$, for all $a \in \mathsf{ind}(\mathcal{A})$, but we strongly conjecture that our results hold without it. The satisfaction relation $\models$ is defined as usual by taking $\mathcal{I} \models C \sqsubseteq D$ iff $C^\mathcal{I} \subseteq D^\mathcal{I}$, $\mathcal{I} \models A(a)$ iff $a \in A^\mathcal{I}$, and $\mathcal{I} \models r(a, b)$ iff $(a, b) \in r^\mathcal{I}$. An interpretation $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$, denoted $\mathcal{I} \models \mathcal{T}$, if $\mathcal{I} \models \alpha$ for all $\alpha \in \mathcal{T}$; it is a model of an ABox $\mathcal{A}$, written $\mathcal{I} \models \mathcal{A}$, if $\mathcal{I} \models \alpha$ for all $\alpha \in \mathcal{A}$; it is a model of a KB $\mathcal{K}$ if $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$.

Query Language. A *positive existential regular path query (PRPQ)* is a formula $\varphi = \exists \mathbf{x}\, \psi(\mathbf{x})$ where ψ is constructed using $\wedge$ and $\vee$ over atoms of the form $\mathcal{E}(t, t')$ where t, t' are variable or constant names, $\mathcal{E}$ is a regular expression over $\{r, r^- \mid r \in \mathsf{N_R}\} \cup \{A? \mid A \in \mathsf{N_C}\}$, and the tuple $\mathbf{x}$ denotes precisely the free variables in ψ. Note that atoms $A(t)$ are captured using $A?(t, t)$.

We denote with I_φ the set of constant names in φ. A *match for φ in $\mathcal{I}$* is a function $\pi : \mathbf{x} \cup I_\varphi \to \Delta^\mathcal{I}$ such that $\pi(a) = a$, for all $a \in I_\varphi$ and $\mathcal{I}, \pi \models \psi(\mathbf{x})$ under the standard semantics of first-order logic extended with the following rule for atoms of the form $\mathcal{E}(t, t')$: $\mathcal{I}, \pi \models \mathcal{E}(t, t')$ if there is a word $\nu_1 \cdots \nu_n \in L(\mathcal{E})$ and a sequence $d_0, \dots, d_n \in \Delta^\mathcal{I}$ such that $d_0 = \pi(t)$, $d_n = \pi(t')$, and for all $i \in [1, n]$ we have that (i) if $\nu_i = A?$, then $d_{i-1} = d_i \in A^\mathcal{I}$, and (ii) if $\nu_i = r$ (resp., $\nu_i = r^-$), then $(d_{i-1}, d_i) \in r^\mathcal{I}$ (resp., $(d_i, d_{i-1}) \in r^\mathcal{I}$). A query φ is *entailed by a KB $\mathcal{K}$*, denoted as $\mathcal{K} \models \varphi$, if there is a match for φ in every model $\mathcal{I}$ of $\mathcal{K}$. The *query entailment problem* asks whether a KB $\mathcal{K}$ entails a PRPQ φ. It is well-known that the *query answering problem* can be reduced to query entailment, and that PRPQs are *preserved under homomorphisms*, that is, if $\mathcal{I} \models \varphi$ and there is a homomorphism from $\mathcal{I}$ to $\mathcal{J}$, then also $\mathcal{J} \models \varphi$.

Additional Notation for Transitive Roles. Given some interpretation $\mathcal{I}$, $\mathcal{I}|_\Delta$ denotes the restriction of $\mathcal{I}$ to domain $\Delta \subseteq \Delta^\mathcal{I}$. For $d \in \Delta^\mathcal{I}$ and $r \in \mathsf{N_R^t}$, the *r-cluster of d in $\mathcal{I}$*, denoted by $Q_{\mathcal{I},r}(d)$, is the set containing d and all elements $e \in \Delta^\mathcal{I}$ such that both $(d, e) \in r^\mathcal{I}$ and $(e, d) \in r^\mathcal{I}$. We call a set $\mathbf{a} \subseteq \Delta^\mathcal{I}$ an *r-cluster in $\mathcal{I}$* if $\mathbf{a} = Q_{\mathcal{I},r}(d)$ for some $d \in \Delta^\mathcal{I}$, and an *r-root cluster* if additionally $(d, e) \in r^\mathcal{I}$ for all $d \in \mathbf{a}$ and $e \in \Delta^\mathcal{I} \setminus \mathbf{a}$. Note that both a single element without an r-loop and a single element with an r-loop are r-clusters of size 1; otherwise r-clusters can be viewed as r-cliques.

3 Tree Decompositions

Existing algorithms for QA in expressive DLs, e.g., $\mathcal{SHIQ}$ (without number restrictions on transitive roles), exploit the fact that for answering queries it suffices to consider *canonical models* that are forest-like, roughly consisting of an interpretation of the ABox and a collection of tree-shaped interpretations whose roots are elements of the ABox. We start with showing that for $\mathcal{SQ}$ this tree-model property is lost.

Example 1. *The number restrictions in $\mathcal{T}$, cf. Section 1, force that every model of $\mathcal{T}$ satisfying* Heart *contains the structure in Fig. 1(a). Moreover, in $\mathcal{SQ}$ clusters can be enforced. Let $\mathcal{T}'$ be the following TBox, where $r \in \mathsf{N_R^t}$:*

$$\{A \sqsubseteq (= 3\, r.B), B \sqsubseteq (= 3\, r.B), A \sqsubseteq \neg B\}.$$

Then, in every model of $\mathcal{T}'$, an element satisfying A roots the structure depicted in Fig. 1(b), where the elements satisfying B form an r-cluster.

Nevertheless, we will establish a *tree-like* model property for $\mathcal{SQ}$, showing that it suffices to consider such models for query entailment. We first introduce a basic form of tree decompositions suited for transitive roles. A *tree* is a prefix-closed subset $T \subseteq (\mathbb{N} \setminus \{0\})^*$. A node $w \in T$ is a successor

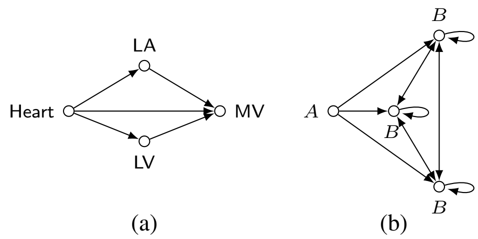

Figure 1: Example 1

of $v \in T$ and v is a predecessor of w if $w = v \cdot i$ for some $i \in \mathbb{N}$. We denote with $w \cdot -1$ the predecessor of w, if it exists.

Definition 1. *A* tree decomposition of an interpretation $\mathcal{I}$ *is pair* $(T, \mathfrak{I})$ *where* T *is a tree and* $\mathfrak{I}$ *is a function that assigns an interpretation* $\mathfrak{I}(w) = (\Delta_w, \cdot^{\mathfrak{I}(w)})$ *to every* $w \in T$, *and the following conditions are satisfied:*

1. $\Delta^{\mathcal{I}} = \bigcup_{w \in T} \Delta_w$;
2. *for every* $w \in T$, *we have* $\mathfrak{I}(w) = \mathcal{I}|_{\Delta_w}$;
3. $r^{\mathcal{I}} = \chi_r$ *for* $r \in \mathsf{N}_\mathsf{R}^{nt}$ *and* $r^{\mathcal{I}} = \chi_r^+$ *for* $r \in \mathsf{N}_\mathsf{R}^{t}$, *where* $$\chi_r = \bigcup_{w \in T} r^{\mathfrak{I}(w)};$$
4. *for every* $d \in \Delta^{\mathcal{I}}$, *the set* $\{w \in T \mid d \in \Delta_w\}$ *is connected in* T.

The width *of* $(T, \mathfrak{I})$ *is the maximum domain size of interpretations that occur in the range of* $\mathfrak{I}$ *minus 1, that is,* $\sup_{w \in T} |\Delta_w| - 1$. *Its* outdegree *is the outdegree of* T.

Unfortunately, this basic tree decomposition does not yet enable tree automata to count over transitive roles (with a small number of states) since the r-successors of an element, say $d \in \Delta^{\mathcal{I}}$, are scattered in the decomposition; see Section 4.1 for further details. To address this, we extend tree decompositions with a third component $\mathfrak{r}$ which assigns to every node $w \in T \setminus \{\varepsilon\}$ a role name $\mathfrak{r}(w)$ and $\bot$ to the root ε. Intuitively, a node labeled with $r = \mathfrak{r}(w)$ is responsible for capturing r-successors of some element(s) in the predecessor of w.

We need some additional notation. Let $(T, \mathfrak{I}, \mathfrak{r})$ be such an extended tree decomposition, and let $w \in T$ and $r \in \mathsf{N}_\mathsf{R}$. We say that $d \in \Delta_w$ *is fresh in* w if $w = \varepsilon$ or $d \notin \Delta_{w \cdot -1}$, and r-*fresh in* w if $r = \mathfrak{r}(w)$ and it is either fresh or $r \neq \mathfrak{r}(w \cdot -1)$. We denote with $F(w)$ and $F_r(w)$ the set of all fresh and r-fresh elements in w, respectively. Intuitively, $F_r(w)$ contains all elements which are allowed to have fresh r-successors in the successor nodes of w. Indeed, the following stronger form of tree decompositions implies (among other things) that, for all d and r, there is a unique w with $d \in F_r(w)$.

Definition 2. *An extended tree decomposition* $\mathfrak{T} = (T, \mathfrak{I}, \mathfrak{r})$ *of an interpretation* $\mathcal{I}$ *is* canonical *if the following conditions are satisfied for every* $w \in T$ *with* $r = \mathfrak{r}(w)$ *and every successor* v *of* w *with* $s = \mathfrak{r}(v)$:

(C$_1$) *if* $(d, e) \in s_1^{\mathfrak{I}(v)}$, *then* $s_1 = s$, *or* $d = e$ *and* $s_1 \in \mathsf{N}_\mathsf{R}^t$;

(C$_2$) *if* $s \in \mathsf{N}_\mathsf{R}^{nt}$, *then* $\Delta_v = \{d, e\}$, *for some* $d \in F(w)$, $e \in F(v)$, *and* $s^{\mathfrak{I}(v)} = \{(d, e)\}$;

(C$_3$) *if* $s \in \mathsf{N}_\mathsf{R}^t$ *and* $r \notin \{\bot, s\}$, *there are* $d \in F(w)$ *and an* r-*root cluster* $\mathbf{a}$ *in* $\mathfrak{I}(v)$ *such that* $\Delta_w \cap \Delta_v = \{d\}$ *and* $d \in \mathbf{a}$; *moreover, there is no successor* $v' \neq v$ *of* w *satisfying this for* d *and* $\mathfrak{r}(v') = s$;

(C$_4$) *if* $s \in \mathsf{N}_\mathsf{R}^t$ *and* $r \in \{\bot, s\}$, *then there is an* s-*root cluster* $\mathbf{a}$ *in* $\mathfrak{I}(v)$ *with:*

- *(a)* $\mathbf{a} \subseteq F_s(w)$;
- *(b)* $\mathbf{a}$ *is an* s-*cluster in* $\mathfrak{I}(w)$;
- *(c)* *for all* $d \in \mathbf{a}$ *and* $(d, e) \in s^{\mathfrak{I}(w)}$, *we have* $e \in \Delta_v$;
- *(d)* *for all* $(d, e) \in s^{\mathfrak{I}(v)}$, $d \in \mathbf{a} \cup F(v)$ *or* $e \notin F(v)$.

Definition 2 imposes restrictions on the structural relation between interpretations at neighboring nodes. Condition (C$_1$) expresses that the interpretation at a node labeled with $\mathfrak{r}(w) = r$ interprets essentially only r non-empty (among role names). Condition (C$_2$) is in analogy with standard unravelling over non-transitive roles (Baader et al. 2003). Condition (C$_3$) reflects that interpretations at neighboring nodes with different $\mathfrak{r}$-components do only interact via single elements. Most interestingly, Condition (C$_4$) plays the role of (C$_2$), but for transitive roles. Note that (C$_4$) is based on r-clusters since they can be enforced, see Example 1 above.

3.1 Tree-like Model Property for $\mathcal{SQ}$

As our first main result, we show a *tree-like model property*, in particular, that every model can be *unraveled* into a canonical decomposition of small width. The proof is via a novel unraveling operation tailored for the logic $\mathcal{SQ}$ and canonical decompositions.

Theorem 1. *Let* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *be an* $\mathcal{SQ}$ *KB and* φ *a PRPQ with* $\mathcal{K} \not\models \varphi$. *There is a model* $\mathcal{J}$ *of* $\mathcal{K}$ *and a canonical tree decomposition* $(T, \mathfrak{I}, \mathfrak{r})$ *of* $\mathcal{J}$ *with (i)* $\mathcal{J} \not\models \varphi$, *(ii)* $\mathfrak{I}(\varepsilon) \models \mathcal{A}$, *and (iii) width and outdegree of* $(T, \mathfrak{I})$ *are bounded by* $O(|\mathcal{A}| \cdot 2^{p(|\mathcal{T}|)})$, *for some polynomial* p.

Before outlining the proof of Theorem 1, we introduce some additional notation. The *width of an interpretation* $\mathcal{I}$ is the minimum k such that $|Q_{\mathcal{I},r}(d)| \leq k$ for all $d \in \Delta^{\mathcal{I}}$, $r \in \mathsf{N}_\mathsf{R}^t$. Moreover, for a transitive role r, we say that e is a *direct* r-*successor of* d if $(d, e) \in r^{\mathcal{I}}$ but $e \notin Q_{\mathcal{I},r}(d)$, and for each f with $(d, f), (f, e) \in r^{\mathcal{I}}$, we have $f \in Q_{\mathcal{I},r}(d)$ or $f \in Q_{\mathcal{I},r}(e)$; if r is non-transitive, then e is a *direct* r-*successor of* d if $(d, e) \in r^{\mathcal{I}}$. The *breadth of* $\mathcal{I}$ is the maximum k such that there are $d, d_1, \ldots, d_k$ and a role name r, all d_i are direct r-successors of d, and

- if r is non-transitive, then $d_i \neq d_j$ for all $i \neq j$;
- if r is transitive, then $Q_{\mathcal{I},r}(d_i) \neq Q_{\mathcal{I},r}(d_j)$, for $i \neq j$.

Let now be $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \not\models \varphi$. As PRPQs are preserved under homomorphisms, the following lemma implies that we can assume without loss of generality that $\mathcal{I}$ is of bounded width and breadth. The proof of this lemma adapts a result in (Kazakov and Pratt-Hartmann 2009).

Lemma 1. *For each* $\mathcal{I} \models \mathcal{K}$, *there is a sub-interpretation* $\mathcal{I}'$ *of* $\mathcal{I}$ *with* $\mathcal{I}' \models \mathcal{K}$ *and width and breadth of* $\mathcal{I}'$ *are bounded by* $O(|\mathcal{A}| + 2^{p(|\mathcal{T}|)})$.

Let $\mathsf{cl}(\mathcal{T})$ be the set of all subconcepts occurring in $\mathcal{T}$, closed under single negation. For each transitive role r, define a binary relation $\rightsquigarrow_{\mathcal{I},r}$ on $\Delta^{\mathcal{I}}$, by taking $d \rightsquigarrow_{\mathcal{I},r} e$ if there is some $(\leqslant n\ r.C) \in \mathcal{T}$ such that $d \in (\leqslant n\ r.C)^{\mathcal{I}}$, $e \in C^{\mathcal{I}}$, and $(d,e) \in r^{\mathcal{I}}$. Based on the transitive, reflexive closure $\rightsquigarrow^*_{\mathcal{I},r}$ of $\rightsquigarrow_{\mathcal{I},r}$, we define, for every $d \in \Delta^{\mathcal{I}}$, the set $\mathsf{Wit}_{\mathcal{I},r}(d)$ of *r-witnesses for d* by:

$$\mathsf{Wit}_{\mathcal{I},r}(d) = \bigcup_{e \mid d \rightsquigarrow^*_{\mathcal{I},r} e} Q_{\mathcal{I},r}(e).$$

Intuitively, $\mathsf{Wit}_{\mathcal{I},r}(d)$ contains all r-witnesses of at-most restrictions of some element d, and due to using $\rightsquigarrow^*_{\mathcal{I},r}$, also all witnesses of at-most restrictions of those witnesses and so on. For the stated bounds, it is important that the size of $\mathsf{Wit}_{\mathcal{I},r}(d)$ is bounded as follows:

Lemma 2. *For every $d \in \Delta^{\mathcal{I}}$ and transitive r, we have $|\mathsf{Wit}_{\mathcal{I},r}(d)| \leq |\mathcal{A}| \cdot 2^{q(|\mathcal{T}|)}$, for some polynomial q.*

We describe now the construction of the interpretation $\mathcal{J}$ and its tree decomposition via a possibly infinite unraveling process. Elements of $\Delta^{\mathcal{J}}$ will be either of the form a with $a \in \mathsf{ind}(\mathcal{A})$ or of the form d_x with $d \in \Delta^{\mathcal{I}}$ and some index x. We usually use δ to refer to domain elements in $\mathcal{J}$ (in either form), and define a function $\tau : \Delta^{\mathcal{J}} \to \Delta^{\mathcal{I}}$ by setting $\tau(\delta) = \delta$, for all $\delta \in \mathsf{ind}(\mathcal{A})$, and $\tau(\delta) = d$, for all δ of the form d_x.

To start the construction of $\mathcal{J}$ and $(T, \mathfrak{I}, \mathfrak{r})$, initialize the domain $\Delta^{\mathcal{J}}$ with $\mathsf{ind}(\mathcal{A}) \cup \bigcup_{r \in \mathsf{N}^t_{\mathsf{R}}} \Delta^r$, where the sets Δ^r are defined as

$$\Delta^r = \{d_r \mid d \in \textstyle\bigcup_{a \in \mathsf{ind}(\mathcal{A})} \mathsf{Wit}_{\mathcal{I},r}(a) \setminus \mathsf{ind}(\mathcal{A})\}.$$

Concept and role names are interpreted in a way such that $\mathcal{J}|_{\mathsf{ind}(\mathcal{A})} = \mathcal{I}|_{\mathsf{ind}(\mathcal{A})}$, and for all $r \in \mathsf{N}^t_{\mathsf{R}}$ and all $\delta, \delta' \in \mathsf{ind}(\mathcal{A}) \cup \Delta^r$, we have

$$\begin{aligned} &\delta \in A^{\mathcal{J}} \Leftrightarrow \tau(\delta) \in A^{\mathcal{I}}, \text{ for all } A \in \mathsf{N_C}, \quad \text{and} \\ &(\delta,\delta') \in r^{\mathcal{J}} \Leftrightarrow (\tau(\delta),\tau(\delta')) \in r^{\mathcal{I}}. \end{aligned} \tag{$\dagger$}$$

Now, initialize $(T, \mathfrak{I}, \mathfrak{r})$ with $T = \{\varepsilon\}$, $\Delta_\varepsilon = \Delta^{\mathcal{J}}$, and $\mathfrak{r}(\varepsilon) = \bot$. This first step ensures that all witnesses of ABox individuals appear in the root.

In the inductive step, we extend $\mathcal{J}$ and $(T, \mathfrak{I}, \mathfrak{r})$ by applying the following rules exhaustively in a fair way.

R$_1$ Let r be non-transitive, $w \in T$, $\delta \in F(w)$, and d a direct r-successor of $\tau(\delta)$ in $\mathcal{I}$ with $\{\delta, d\} \not\subseteq \mathsf{ind}(\mathcal{A})$. Then, add a fresh successor v of w to T, add the fresh element d_v to $\Delta^{\mathcal{J}}$, extend $\mathcal{J}$ by adding $(\delta, d_v) \in r^{\mathcal{J}}$ and $d_v \in A^{\mathcal{J}}$ iff $d \in A^{\mathcal{I}}$, for all $A \in \mathsf{N_C}$, and set $\Delta_v = \{\delta, d_v\}$ and $\mathfrak{r}(v) = r$.

R$_2$ Let r be transitive, $w \in T$, and $\delta_0 \in F(w)$ such that:

(a) $w = \varepsilon$ and $\delta_0 \in \Delta^s$, for some transitive $s \neq r$, or

(b) $w \neq \varepsilon$ and $\mathfrak{r}(w) \neq r$.

Then add a fresh successor v of w to T, and define

$$\Delta = \{e_v \mid e \in \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta_0)) \setminus \{\tau(\delta_0)\}\}.$$

Extend the domain of $\mathcal{J}$ with Δ and the interpretation of concept and role names such that ($\dagger$) is satisfied for all $\delta, \delta' \in \Delta \cup \{\delta_0\}$. Finally, set $\Delta_v = \Delta \cup \{\delta_0\}$ and $\mathfrak{r}(v) = r$.

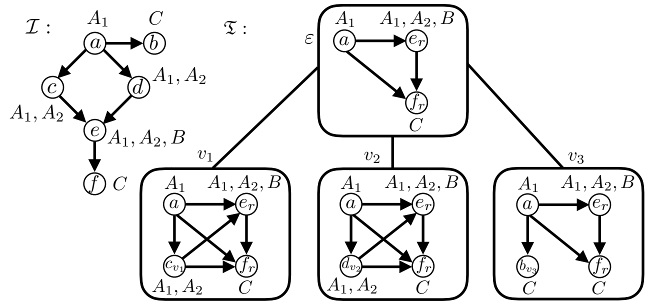

Figure 2: Example 2

R$_3$ Let r be transitive, $w \in T$, $\mathbf{a} \subseteq F_r(w)$ an r-cluster in $\mathfrak{I}(w)$ such that:

(a) $w = \varepsilon$ and $\mathbf{a} \subseteq \Delta^r \cup \mathsf{ind}(\mathcal{A})$, or

(b) $w \neq \varepsilon$ and $\mathfrak{r}(w) = r$.

If there is a direct r-successor e of $\tau(\delta)$ in $\mathcal{I}$ for some $\delta \in \mathbf{a}$ such that $(\delta, \delta') \notin r^{\mathcal{J}}$ for any δ' with $\tau(\delta') = e$, then add a fresh successor v of w to T, and define

$$\Delta = \{f_v \mid f \in \mathsf{Wit}_{\mathcal{I},r}(e) \setminus \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta))\} \quad \text{and}$$

$$\Delta_v = \Delta \cup \mathbf{a} \cup \{\delta'' \mid r(\delta', \delta'') \in \mathfrak{I}(w) \text{ for some } \delta' \in \mathbf{a}\}.$$

Then extend the domain of $\mathcal{J}$ with Δ and the interpretation of concept names such that ($\dagger$) is satisfied for all $\delta \in \mathbf{a} \cup \Delta$ and $\delta' \in \Delta_v$. Finally, set $\mathfrak{r}(v) = r$.

To finish the construction, let $\mathcal{J}$ be the interpretation obtained in the limit, and set $\mathfrak{I}(w) = \mathcal{J}|_{\Delta_w}$, for all $w \in T$. It is verified in the appendix that $(T, \mathfrak{I}, \mathfrak{r})$ and $\mathcal{J}$ satisfy the conditions from Theorem 1. Notably, τ is a homomorphism from $\mathcal{J}$ to $\mathcal{I}$, thus $\mathcal{J} \not\models \varphi$, due to preservation under homomorphisms.

Rules **R**$_1$–**R**$_3$ are, respectively, in one-to-one correspondence with Conditions (C$_2$)–(C$_4$) in Definition 2. In particular, **R**$_1$ implements the well-known unraveling procedure for non-transitive roles. **R**$_2$ is used to change the 'role component' for transitive roles by creating a fresh node whose interpretation contains all witnesses of the chosen element δ. Finally, **R**$_3$ describes how to unravel direct r-successors in case of transitive roles r. In the definition of Δ it is taken care that witnesses which are 'inherited' from predecessors are not introduced again, in order to preserve at-most restrictions.

We finish the section with an illustrating example.

Example 2. *Let $\mathcal{K}$ be the following KB, where $r \in \mathsf{N}^t_{\mathsf{R}}$:*

$$(\{A_1 \sqsubseteq (\leqslant 1\ r.B), A_2 \sqsubseteq (\leqslant 1\ r.C)\}, \{A_1(a)\}).$$

*Figure 2 shows a model $\mathcal{I}$ of $\mathcal{K}$ and a canonical decomposition $\mathfrak{T}$ of its unraveling (transitivity connections are omitted). In the initialization phase, the interpretation $\mathfrak{I}(\varepsilon)$ is constructed starting from individual a. Since $a \rightsquigarrow_{\mathcal{I},r} e$ and $e \rightsquigarrow_{\mathcal{I},r} f$, we have $\mathsf{Wit}_{\mathcal{I},r}(a) = \{e, f\}$, thus e_r and f_r are added in this phase. The interpretations $\mathfrak{I}(v_i)$ are introduced using **R**$_3$: In all cases Δ_ε is the cluster $\mathbf{a}$ and $\delta = a$; and, e.g., $\Delta = \{c_{v_1}\}$ for $\mathfrak{I}(v_1)$.*

4 Automata-Based Query Entailment

In this section, we devise an automata-based decision procedure for query entailment in $\mathcal{SQ}$. We start with the necessary background about the used automata model.

Alternating Tree Automata. A tree is *k-ary* if each node has *exactly* k successors. For brevity, we set $[k] = \{-1, 0, \dots, k\}$. Let Σ be a finite alphabet. A Σ*-labeled tree* is a pair (T, τ) with T a tree and $\tau : T \to \Sigma$ assigns a letter from Σ to each node. A *two-way alternating tree automaton (2ATA)* over Σ-labeled k-ary trees is a tuple $\mathfrak{A} = (Q, \Sigma, q_0, \delta, F)$ where Q is a finite set of *states*, $q_0 \in Q$ is an *initial state*, δ is the *transition function*, and F is the *(parity) acceptance condition* (Vardi 1998). The transition function maps a state q and an input letter $a \in \Sigma$ to a positive Boolean formula over the constants true and false, and variables from $[k] \times Q$. The semantics is given in terms of *runs*, see appendix. As usual, $L(\mathfrak{A})$ denotes the set of trees accepted by $\mathfrak{A}$. Emptiness of $L(\mathfrak{A})$ can be checked in exponential time in the number of states of $\mathfrak{A}$ (Vardi 1998).

General Picture. The leading thought is as follows. If $\mathcal{K} \not\models \varphi$, then, by Theorem 1, there is a model $\mathcal{J}$ of $\mathcal{K}$ and a canonical tree decomposition thereof with small width and outdegree such that $\mathcal{J} \not\models \varphi$. The idea is to design 2ATAs $\mathfrak{A}_{\mathsf{can}}$, $\mathfrak{A}_{\mathcal{K}}$, and $\mathfrak{A}_{\varphi}$ which accept canonical tree decompositions, (tree-like) models of the KB $\mathcal{K}$, and (tree-like) models of the query φ, respectively. Query answering is then reduced to the question whether some tree is accepted by $\mathfrak{A}_{\mathsf{can}}$ and $\mathfrak{A}_{\mathcal{K}}$, but not by $\mathfrak{A}_{\varphi}$. As we shall see, these automata have size exponential in $\mathcal{K}$ and can be constructed in double exponential time. Since 2ATAs can be complemented and intersected in polynomial time, the automaton $\mathfrak{A}_{\mathsf{can}} \wedge \mathfrak{A}_{\mathcal{K}} \wedge \neg\mathfrak{A}_{\varphi}$ is of exponential size, and can be constructed in double exponential time. Checking it for nonemptiness can thus be done in double exponential time. A matching lower bound is inherited from positive existential query answering in $\mathcal{ALC}$ (Calvanese, Eiter, and Ortiz 2014). We thus obtain our main result.

Theorem 2. *PRPQ entailment over $\mathcal{SQ}$-knowledge bases is* 2EXPTIME*-complete.*

Encoding Tree Decompositions. As the underlying interpretation might be infinite, 2ATAs cannot directly work over tree decompositions. Thus, for the desired approach to work, it is crucial to *encode* tree decompositions using a finite alphabet. To this aim, we use an approach similar to (Grädel and Walukiewicz 1999).

Throughout this section, fix a knowledge base $\mathcal{K}$ and let K and k be the bounds on width and outdegree, respectively, obtained in Theorem 1. Then, fix a finite set Δ having $2K$ elements with $\mathsf{ind}(\mathcal{A}) \subseteq \Delta$, and define $\Sigma = \{\bullet\} \cup \Sigma'$, where Σ' is the set of all pairs $(\mathcal{I}, x)$ such that $\mathcal{I}$ is an interpretation where only symbols from $\mathcal{K}$ are interpreted non-empty, $\Delta^{\mathcal{I}} \subseteq \Delta$, $|\Delta^{\mathcal{I}}| \leq K$, and x is either a role name from $\mathcal{K}$ or $\perp$. The symbol $\bullet \in \Sigma$ is used to encode non-existing branches (tree decompositions are not necessarily uniformly branching).

Let (T, τ) be a Σ-labeled tree with Σ as above. For convenience, we use $\mathcal{I}_w$ and r_w to refer to the single components of τ in a node w with $\tau(w) \neq \bullet$, that is, $\tau(w) = (\mathcal{I}_w, r_w)$. Given an element $d \in \Delta$, we say that $v, w \in T$ are d*-connected* iff $d \in \Delta^{\mathcal{I}_u}$ for all u on the unique shortest path from v to w. In case $d \in \Delta^{\mathcal{I}_w}$, we use $[w]_d$ to denote the set of all v which are d-connected to w. We call (T, τ) *consistent* if ε is the only node with $r_\varepsilon = \perp$ and $(\mathcal{I}_w)|_D = (\mathcal{I}_v)|_D$ for all neighbors $v, w \in T$ and $D = \Delta^{\mathcal{I}_w} \cap \Delta^{\mathcal{I}_v}$. A consistent Σ-labeled tree (T, τ) *represents* a triple $(T, \mathfrak{I}, \mathfrak{r})$ of width at most K as follows. The domain underlying $(T, \mathfrak{I}, \mathfrak{r})$ is the set of all elements $[w]_d$ with $w \in T$ and $d \in \Delta^{\mathcal{I}_w}$, and for every $w \in T$, the interpretation $\mathfrak{I}(w)$ is defined as:

$$\Delta_w = \{[w]_d \mid d \in \Delta^{\mathcal{I}_w}\}, \quad A^{\mathfrak{I}(w)} = \{[w]_d \mid d \in A^{\mathcal{I}_w}\},$$
$$r^{\mathfrak{I}(w)} = \{([w]_d, [w]_e) \mid (d, e) \in r^{\mathcal{I}_w}\},$$

for all concept names A and role names r occurring in $\mathcal{K}$; and $\mathfrak{r}(w)$ is just r_w. We denote with $\mathcal{I}_{(T,\tau)}$ the interpretation $\bigcup_{w \in T} \mathcal{I}_w$; clearly, $(T, \mathfrak{I}, \mathfrak{r})$ is a tree decomposition of $\mathcal{I}_{(T,\tau)}$. As a convention, we use $[\varepsilon]_a$ to represent each ABox individual $a \in \mathsf{ind}(\mathcal{A})$ in the encoding. Based on the size $2K$ of Δ, it is not hard to verify that, conversely, for every width K tree decomposition of some $\mathcal{I}$, there is a consistent (T, τ) such that $\mathcal{I}_{(T,\tau)}$ is isomorphic to $\mathcal{I}$.

It is easy to devise a 2ATA $\mathfrak{A}_{\mathsf{can}}$ which accepts an input (T, τ) iff it is consistent and the represented tree decomposition $(T, \mathfrak{I}, \mathfrak{r})$ is canonical. We thus concentrate on the most challenging automata $\mathfrak{A}_{\mathcal{K}}$ and $\mathfrak{A}_{\varphi}$.

4.1 Knowledge Base Automaton $\mathfrak{A}_{\mathcal{K}}$

The automaton $\mathfrak{A}_{\mathcal{K}}$ is the intersection of two automata $\mathfrak{A}_{\mathcal{A}}$ and $\mathfrak{A}_{\mathcal{T}}$ verifying that the input satisfies the ABox and the TBox, respectively. Note that, by Point (ii) of Theorem 1, we can assume that the ABox is satisfied in the root; thus, an automaton $\mathfrak{A}_{\mathcal{A}}$ checking whether $\mathcal{I}_{(T,\tau)} \models \mathcal{A}$ just has to check the label $\tau(\varepsilon)$, see the appendix.

For the design of the automaton $\mathfrak{A}_{\mathcal{T}}$, assume w.l.o.g. that $\mathcal{T}$ is of the form $\{\top \sqsubseteq C_{\mathcal{T}}\}$ and $C_{\mathcal{T}}$ is in negation normal form. We present the main ideas of the construction of $\mathfrak{A}_{\mathcal{T}}$, see the appendix for further details. In its 'outer loop', the automaton visits every domain element d in state $C_{\mathcal{T}}(d)$. This is realized using the initial state q_0, and states of the form $D(d)$, D a sub-concept of $C_{\mathcal{T}}$ and $d \in \Delta$ via the following transitions for every $(\mathcal{I}, x) \in \Sigma$:

$$\delta(q_0, (\mathcal{I}, x)) = \textstyle\bigwedge_{1 \leq i \leq K} (i, q_0) \wedge \bigwedge_{d \in \Delta^{\mathcal{I}}} (0, C_{\mathcal{T}}(d))$$
$$\delta(q_0, \bullet) = \mathsf{true}$$

If $\mathfrak{A}_{\mathcal{T}}$ visits w in a state $D(d)$ this presents the obligation to verify that, in the represented model, $[w]_d$ satisfies D. The Boolean operations are dealt with using the following transitions, for every $(\mathcal{I}, x) \in \Sigma$:

$$\delta(A(d), (\mathcal{I}, x)) = \text{if } d \in A^{\mathcal{I}}, \text{then } \mathsf{true} \text{ else } \mathsf{false}$$
$$\delta(\neg A(d), (\mathcal{I}, x)) = \text{if } d \notin A^{\mathcal{I}}, \text{then } \mathsf{true} \text{ else } \mathsf{false}$$
$$\delta((C_1 \sqcup C_2)(d), (\mathcal{I}, x)) = (0, C_1(d)) \vee (0, C_2(d))$$
$$\delta((C_1 \sqcap C_2)(d), (\mathcal{I}, x)) = (0, C_1(d)) \wedge (0, C_2(d))$$

For states of the form $(\sim n\ r.D)(d)$ we have to be more careful. The naive approach for counting the number of r-successors of d satisfying D would be to count the number of r-successors satisfying D in the interpretation associated to the current node, and then move to all other nodes where d appears. Since interpretations associated to neighboring nodes might overlap, to avoid double counting, we have to store (in the states) all elements that have already been counted in the current node before changing the node. However, since the domain in each node has size exponential in $|\mathcal{T}|$, we need doubly exponentially many states for this task. Since this naive approach does not result in optimal complexity, we pursue an alternative approach, based on canonicity, leading to only exponentially many states.

Our approach is based on characterizing how r-successors of an element can be uniquely identified in canonical tree decompositions. Assume some $(T,\tau) \in L(\mathfrak{A}_{\mathsf{can}})$ and let r be a role name. In what follows, we assume that the notions of 'fresh' and 'r-fresh' are lifted to the encoding in the straightforward way. An *r-path from $[w]_d$ to $[v]_e$ in (T,τ)* is a sequence $d_0, w_0, d_1, \ldots, w_{n-1}, d_n$ such that $d = d_0$, $e = d_n$, $w_0 \in [w]_d$, $w_{n-1} \in [v]_e$, and $(d_i, d_{i+1}) \in r^{\mathcal{I}_{w_i}}$, for all $0 \le i < n$. It is *downward* if, for all $0 < i < n$, w_i is a successor of w_{i-1} and d_i is contained in an r-root cluster of w_i. We then have:

Lemma 3. *For $(T,\tau) \in L(\mathfrak{A}_{\mathsf{can}})$, we have $([w]_d, [v]_e) \in r^{\mathcal{I}_{(T,\tau)}}$ iff one of the following is true:*

- *r is non-transitive and $(d,e) \in r^{\mathcal{I}_\varepsilon}$ or $(d,e) \in r^{\mathcal{I}_v}$, d is fresh in w, and v is a successor of w, or*
- *r is transitive, and there is an r-path $d_0, w_0, \ldots, d_n$ from $[w]_d$ to $[v]_e$ such that one of the following holds:*
  - ***A*** *$d_0 \in F_r(w_0) \cup F_r(w_0 \cdot -1)$, $d_1 \in F_r(w_0)$, and $d_0, \ldots, d_n$ is downward, or*
  - ***B*** *$d_0 \in F_r(w_0)$, $d_1 \notin F_r(w_0)$, and if $n > 1$, then $d_1, \ldots, d_n$ is downward and $w_1 \cdot -1 \in [w]_{d_1}$ is an ancestor of w_0 such that $d_1 \in F_r(w_1 \cdot -1)$.*

This lemma suggests the following approach for verifying the obligation $(\sim n\ r.D)(d)$ at some node w. If r is non-transitive, 'navigate' with the automaton to the (unique!) w^* such that $d \in F(w^*)$ and count the r-successors of d in the successors v of w^*, or in ε. If r is transitive, navigate with the automaton to the unique w^* such that $d \in F_r(w^*)$ and change to a state $q^*_{(\sim n\ r.D),d}$, starting from which $\mathfrak{A}_\mathcal{T}$ systematically scans the r-successors according to **A** and **B**. We concentrate on verifying at-least restrictions, at-most restrictions are completely complementary.

Assume $\tau(w^*) = (\mathcal{I}, x)$, and let $\mathbf{a}_1, \ldots, \mathbf{a}_\ell$ be all r-clusters in $\mathcal{I}$ reachable from d (including $Q_{\mathcal{I},r}(d)$), and let $a_1, \ldots, a_\ell$ be representatives of these clusters. Moreover, let N be the set of all tuples $\mathbf{n} = (n_1, \ldots, n_\ell)$ such that $\sum_i n_i = n$. Then, the transition $\delta(q^*_{(\geqslant n\ r.D),d}, (\mathcal{I}, x))$ is defined as

$$\bigvee_{\mathbf{n}\in N} \bigvee_{X \subseteq [1,\ell]} \bigwedge_{i \in X} (0, q^{\mathbf{A}}_{(\geqslant n_i\ r.D),a_i}) \wedge \bigwedge_{i \in [1,\ell]\setminus X} (0, q^{\mathbf{B}}_{(\geqslant n_i\ r.D),a_i}).$$

Thus, $\mathfrak{A}_\mathcal{T}$ guesses a distribution of n to the reachable clusters. Moreover, it guesses from which clusters it starts paths of the shape **A** and **B**. For both guesses, it verifies that the chosen a_i is r-fresh (for **A**) or not (for **B**), and continues in states $q^{\downarrow}_{(\geqslant n\ r.D)}$ and $q^{\uparrow}_{(\geqslant n\ r.D)}$, respectively. This is done using the following transitions:

$$\delta(q^{\mathbf{A}}_{(\geqslant n\ r.D),d}, (\mathcal{I}, x)) = (0, F_{r,d}) \wedge (0, q^{\downarrow}_{(\geqslant n\ r.D),d})$$

$$\delta(q^{\mathbf{B}}_{(\geqslant n\ r.D),d}, (\mathcal{I}, x)) = (0, \overline{F}_{r,d}) \wedge (-1, q^{\uparrow}_{(\geqslant n\ r.D),d})$$

$$\delta(F_{r,d}, (\mathcal{I}, \bot)) = \mathsf{true}$$

$$\delta(F_{r,d}, (\mathcal{I}, x)) = \mathsf{false} \qquad \text{if } x \notin \{r, \bot\}$$

$$\delta(F_{r,d}, (\mathcal{I}, r)) = (-1, F'_{r,d})$$

$$\delta(F'_{r,d}, (\mathcal{I}, x)) = \begin{cases} \mathsf{true} & \text{if } x \notin \{r,\bot\} \text{ or } d \notin \Delta^\mathcal{I}, \\ \mathsf{false} & \text{otherwise,} \end{cases}$$

and complementary transitions for $\overline{F}_{r,d}$. Now, in states $q^{\uparrow}_{(\geqslant n\ r.D),d}$, the automaton goes up until it finds the world where d is r-fresh (corresponding to $w_1 \cdot -1$ in **B**) and looks for downward paths starting from there. This is done by taking setting $\delta(q^{\uparrow}_{(\geqslant n\ r.D),d}, (\mathcal{I}, x)) = \mathsf{false}$ whenever $d \notin \Delta^\mathcal{I}$, and otherwise:

$$\delta(q^{\uparrow}_{(\geqslant n\ r.D),d}, (\mathcal{I}, x)) = (0, q^{\mathbf{A}}_{(\geqslant n\ r.D),d}) \vee (0, q^{\mathbf{B}}_{(\geqslant n\ r.D),d}).$$

It thus remains to describe transitions for states of the form $q^{\downarrow}_{(\geqslant n\ r.D),d}$ at some node w. Such situations represent the obligation to find n r-successors along downward paths from d. Note that the transitions before ensure that $d \in F_r(w)$. In this case, the automaton guesses how many of the n successors it will find locally in the current cluster (using states $p^{\mathsf{loc}}_{m,r,D,d}$), and how many are to be found in successor nodes (using $p^{\mathsf{succ}}_{(\geqslant m\ r.D)}$). Formally, let M be the set of all tuples $\mathbf{m} = (m_0, \ldots, m_k)$ with $\sum_i m_i = n$, and define the transition for $\delta(q^{\downarrow}_{(\geqslant n\ r.D),d}, (\mathcal{I}, x))$ as:

$$\bigvee_{\mathbf{m}\in M} \Big((0, p^{\mathsf{loc}}_{m_0,r,D,d}) \wedge \bigwedge_{i \in [1,k]} (i, p^{\mathsf{succ}}_{(\geqslant m_i\ r.D),d}) \Big)$$

States of the form $p^{\mathsf{loc}}_{n,r,D,d}$ are used to verify that in $Q_{\mathcal{I},r}(d)$ there are n elements satisfying D:

$$\delta(p^{\mathsf{loc}}_{n,r,D,d}, (\mathcal{I}, x)) = \bigvee_{Y \subseteq Q_{\mathcal{I},r}(d), |Y| = n} \bigwedge_{e \in Y} D(e).$$

It remains to give the transitions for states $p^{\mathsf{succ}}_{(\geqslant m\ r.D)}$. To start, we set $\delta(p^{\mathsf{succ}}_{(\geqslant n\ r.D),d}, \sigma) = \mathsf{true}$, whenever $n = 0$; $\delta(p^{\mathsf{succ}}_{(\geqslant n\ r.D),d}, \bullet) = \mathsf{false}$; and $\delta(p^{\mathsf{succ}}_{(\geqslant n\ r.D),d}, (\mathcal{I}, x)) = \mathsf{false}$ whenever $x \neq r$ or d is not in a root cluster of $\mathcal{I}$. For all other cases, let $\mathbf{a}_1, \ldots, \mathbf{a}_\ell$ be all r-clusters reachable from d, except $Q_{\mathcal{I},r}(d)$, let N be again the set of all $\mathbf{n} = (n_1, \ldots, n_\ell)$ such that $\sum_i n_i = n$, and include the transition

$$\delta(p^{\mathsf{succ}}_{(\geqslant n\ r.D),d}, (\mathcal{I}, x)) = \bigvee_{\mathbf{n}\in N} \bigwedge_{i \in [1,\ell]} (0, q^{\mathbf{A}}_{(\geqslant n_i\ r.D),a_i}).$$

Using the parity condition, we make sure that states $q^{\downarrow}_{(\geqslant n\ r.D),d}$ with $n \ge 1$ are not suspended forever, that is, eventualities are finally satisfied.

Lemma 4. *For every $(T,\tau) \in L(\mathfrak{A}_{\mathsf{can}})$, we have $(T,\tau) \in L(\mathfrak{A}_\mathcal{T})$ iff $\mathcal{I}_{(T,\tau)} \models \mathcal{T}$. It can be constructed in time double exponential in $|\mathcal{K}|$, and has exponentially many states in $|\mathcal{K}|$.*

4.2 Query Automaton $\mathfrak{A}_\varphi$

In previous work, we have observed that the approach for the query automaton taken in (Calvanese, Eiter, and Ortiz 2014) leads to a 2ATA with double exponentially many states in $\mathcal{K}$, and thus not to optimal complexity (Gutiérrez-Basulto, Ibáñez-García, and Jung 2017b). We thus take an alternative approach by first giving an intermediate characterization for when a query has a match, and then show how to exploit this to build a 2ATA with exponentially many states.

Fix a P2RPQ $\varphi = \exists \mathbf{x}\, \psi(\mathbf{x})$. Note first that since for every regular expression $\mathcal{E}$ over some alphabet Γ, one can construct in polynomial time an equivalent non-deterministic finite automaton (NFA) $\mathfrak{B} = (Q_\mathfrak{B}, \Gamma, s_{0\mathfrak{B}}, \Delta_\mathfrak{B}, F_\mathfrak{B})$ (Fürer 1980), we generally assume an NFA-based representation, that is, atoms in φ take the shape $\mathfrak{B}(t, t')$, $\mathfrak{B}$ an NFA. For states $s, s' \in Q_\mathfrak{B}$, write $\mathfrak{B}_{s,s'}$ for the NFA that is obtained from $\mathfrak{B}$ by taking s as initial state and $\{s'\}$ as the set of final states. To give semantics to the automata based representation, we define $\mathcal{I} \models \mathfrak{B}(a, b)$ iff $\mathcal{I} \models \mathcal{E}_\mathfrak{B}(a, b)$, where $\mathcal{E}_\mathfrak{B}$ is a regular expression equivalent to $\mathfrak{B}$.

A *conjunctive regular path query (CRPQ)* is a PRPQ which does not use $\vee$. It is well-known that the PRPQ φ is equivalent to a disjunction $q_1 \vee \ldots \vee q_n$ of CRPQs, where n is exponential in $|\varphi|$. Given a CRPQ p, we denote with $\hat{p}$ the equivalent CRPQ obtained from p by replacing every occurrence of r or r^-, r transitive, with $r \cdot r^*$ or $r^- \cdot (r^-)^*$, respectively. Let (T, τ) be a consistent Σ-labeled tree. In the appendix, we show the following characterization.

Lemma 5. *A function $\pi : \mathbf{x} \cup I_\varphi \to \Delta^{\mathcal{I}_{(T,\tau)}}$ with $\pi(a) = [\varepsilon]_a$, for every $a \in I_\varphi$, is a match for φ in $\mathcal{I}_{(T,\tau)}$ iff there is a q_i such that for every $\mathfrak{B}(t, t')$ in $\hat{q}_i$, there is a sequence*

$$(d_0, s_0), w_1, (d_1, s_1), w_2, \ldots, w_n, (d_n, s_n),$$

where $(d_i, s_i) \in \Delta \times Q_\mathfrak{B}$ and $w_i \in T$ and such that:

(a) $s_0 = s_{0\mathfrak{B}}$, $s_n \in F_\mathfrak{B}$,

(b) $\pi(t) = [w_1]_{d_0}$, $\pi(t') = [w_n]_{d_n}$, *and*

(c) for every $i \in [1, n]$, we have $d_{i-1}, d_i \in \Delta^{\mathcal{I}_{w_i}}$, $w_i \in [w_{i-1}]_{d_{i-1}}$ if $i > 1$, and $\mathcal{I}_{w_i} \models \mathfrak{B}_{s_{i-1}, s_i}(d_{i-1}, d_i)$.

We will refer to such sequences as *witness sequences*. The lemma suggests the following approach. In order to check whether φ has a match in $\mathcal{I}_{(T,\tau)}$, the automaton guesses a q_i and tries to find the witness sequences characterizing a match. For this purpose, $\mathfrak{A}_\varphi$ uses as states triples $\langle p, V_l, V_r \rangle$ such that $p \subseteq \hat{q}_i$, $I_p = \emptyset$, and:

- V_l and V_r are sets of expressions of the form $(d, s) \to_\mathfrak{B} x$ and $x \to_\mathfrak{B} (d, s)$, respectively, where $\mathfrak{B}$ is the automaton of some atom $\mathfrak{B}(t, t')$ in $\hat{q}_i$, $s \in Q_\mathfrak{B}$, $d \in \Delta$, $x \in \mathsf{var}(p)$.

Intuitively, when the automaton visits a node w in state $\langle p, V_l, V_r \rangle$, this represents the obligation that each atom $\mathfrak{B}(x, y)$ in p still has to be processed in the sense that all variables occuring in p will be instantiated in the subtree rooted at w, and

- for each $(d, s) \to_\mathfrak{B} x \in V_l$, $\mathfrak{A}_\varphi$ tries to find a suffix of the witness sequence for $\mathfrak{B}(t, t')$ starting with (d, s),
- for each $x \to_\mathfrak{B} (d, s) \in V_r$, $\mathfrak{A}_\varphi$ tries to find a prefix of the witness sequence for $\mathfrak{B}(t, t')$ ending with (d, s).

We describe verbally how the automaton $\mathfrak{A}_\varphi$ acts when visiting a node w in state $\langle p, V_l, V_r \rangle$; the complete transition function is given in the appendix. First, $\mathfrak{A}_\varphi$ non-deterministically chooses a partition $S_0, \ldots, S_k$ (with S_i possibly empty, for all i) of $\mathsf{var}(p)$ and values $d_x \in \Delta^{\mathcal{I}_w}$ for all $x \in S_0$. Intuitively, S_0 contains the variables that are to be instantiated in w, and S_i contains the variables that are to be instantiated in the subtree rooted at $w \cdot i$. Based on the taken choice, $\mathfrak{A}_\varphi$ determines states $\langle p^i, V_l^i, V_r^i \rangle$ which are then sent to the respective successors $i \in [1, k]$ of w. Using the parity condition, we enforce that every variable is instantiated after finitely many of such steps.

We demonstrate on several examples how to compute the states $\langle p^i, V_l^i, V_r^i \rangle$ from $S_0, \ldots, S_k$ and d_x for all $x \in S_0$.

- Assume some $\mathfrak{B}(x, y) \in p$ with $x, y \in S_0$. In this case, $\mathfrak{A}_\varphi$ guesses some $f \in F_\mathfrak{B}$ and verifies (using another set of states) that there is a witness sequence for $\mathfrak{B}(x, y)$ starting with $(d_x, s_{0\mathfrak{B}})$ and ending with (d_y, s_f).
- Assume $\mathfrak{B}(x, y) \in p$ and $x, y \in S_i$ for some $i > 0$. In this case, just put $\mathfrak{B}(x, y)$ into p^i.
- For an atom $\mathfrak{B}(x, y) \in p$ with $x \in S_0$ and $y \in S_i$ for $i > 0$, $\mathfrak{A}_\varphi$ guesses an intermediate tuple (d, s), verifies that there is a witness sequence from $(d_x, s_{0\mathfrak{B}})$ to (d, s) and adds $x \to_\mathfrak{B} (d, s)$ to V_r^i.
- For the treatment of V_l (V_r is similar), assume $(d, s) \to_\mathfrak{B} x \in V_l$. If $x \in S_0$, $\mathfrak{A}_\varphi$ verifies that the sequence has a suffix from (d, s) to (d_x, s_f), for some $s_f \in F_\mathfrak{B}$. If $x \in S_i$, $i > 0$, $\mathfrak{A}_\varphi$ guesses an intermediate pair (d', s'), verifies that there is an infix between (d, s) and (d', s') and includes $(d', s') \to_\mathfrak{B} x \in V_l^i$.

We show in the appendix how to verify the existence of an infix of a witness sequence between two pairs (d, s) and (d', s') as required in the first, third and last item using only exponentially many states. Regarding number of states, observe that there are only exponentially many disjuncts (and thus states) q_i and exponentially many states of the form $\langle p, V_l, V_r \rangle$ as described.

We refer the reader to the appendix for the complete construction and a proof of the following lemma.

Lemma 6. *There is a 2ATA $\mathfrak{A}_\varphi$ such that for every $(T, \tau) \in L(\mathfrak{A}_{\mathsf{can}})$, we have $(T, \tau) \in L(\mathfrak{A}_\varphi)$ iff $\mathcal{I}_{(T,\tau)} \models q$. It can be constructed in exponential time in $|\varphi| + |\mathcal{K}|$ and has exponentially in $|\varphi| + |\mathcal{K}|$ many states.*

5 Discussion and Future Work

The obtained results are both of practical and theoretical interest. From the practical point of view, our complexity results and application demands open up the possibility to include a profile based on $\mathcal{SQ}$ to OWL 2. Note that there is no increase in the computational complexity in comparison with that of $\mathcal{SQ}$ *without* counting over transitive roles. From the theoretical perspective, our techniques are useful for several future lines of research. First, the unraveling lays the groundwork for studying extensions of $\mathcal{SQ}$ with other DL constructors. Second, the technique underlying the query automaton works for standard tree decompositions (it does not rely on canonicity) of bounded outdegree, even if the width is high (exponential in our case). We thus believe that this technique is useful for query answering in other DLs. Finally, the gained understanding of the model-theoretic characteristics of $\mathcal{SQ}$ is an important step towards the development of more practical decision procedures.

As future work, we will tackle the following four interesting problems: (i) The *data complexity* of deciding entailment of PRPQs in $\mathcal{SQ}$. The present techniques give only exponential bounds, but we expect coNP-completeness. (ii) The complexity of deciding entailment of *conjunctive queries (CQs)* in $\mathcal{SQ}$. The proposed automata-based approach yields the same upper bound for PRPQs or CQs, but we expect it to be easier for CQs. (iii) The complexity of deciding query entailment in generalizations of $\mathcal{SQ}$ with *role composition* or regular expressions on roles; or with nominals and (controlled) inverses. (iv) The complexity of query entailment in $\mathcal{SQ}$ over *finite* models. Indeed, $\mathcal{SQ}$ lacks *finite controlability*, that is, query entailment in the finite does not coincide with unrestricted query entailment:

Example 3. *Consider* $\mathcal{A} = \emptyset$, $\mathcal{T} = \{\top \sqsubseteq \exists r.\top\}$, *and* $\varphi = \exists x\, r(x,x)$ *for some* $r \in \mathsf{N}_\mathsf{R}^t$. *Clearly,* $(\mathcal{T},\mathcal{A}) \not\models \varphi$, *but for every finite model* $\mathcal{I}$ *of* $(\mathcal{T},\mathcal{A})$, *we have* $\mathcal{I} \models \varphi$.

Acknowledgments

The first author was funded by EU's Horizon 2020 programme under the Marie Skłodowska-Curie grant 663830, the second one by the FWF project P30360, and the third one by the ERC grant 647289 CODA.

References

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Baget, J.; Bienvenu, M.; Mugnier, M.; and Thomazo, M. 2017. Answering conjunctive regular path queries over guarded existential rules. In *IJCAI*, 793–799. ijcai.org.

Bienvenu, M.; Ortiz, M.; and Simkus, M. 2015. Regular path queries in lightweight description logics: Complexity and algorithms. *J. Artif. Intell. Res. (JAIR)* 53:315–374.

Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Vardi, M. Y. 2000. Containment of conjunctive regular path queries with inverse. In *KR*, 176–185. Morgan Kaufmann.

Calvanese, D.; Eiter, T.; and Ortiz, M. 2009. Regular path queries in expressive description logics with nominals. In *Proc. of IJCAI-09*, 714–720.

Calvanese, D.; Eiter, T.; and Ortiz, M. 2014. Answering regular path queries in expressive description logics via alternating tree-automata. *Inf. Comput.* 237:12–55.

Dogrusoz, U.; Cetintas, A.; Demir, E.; and Babur, O. 2009. Algorithms for effective querying of compound graph-based pathway databases. *BMC Bioinformatics* 10(1):376.

Eiter, T.; Lutz, C.; Ortiz, M.; and Simkus, M. 2009. Query answering in description logics with transitive roles. In *Proc. of IJCAI-09*, 759–764.

Florescu, D.; Levy, A. Y.; and Suciu, D. 1998. Query containment for conjunctive queries with regular expressions. In *Proc. of PODS*, 139–148. ACM Press.

Fürer, M. 1980. The complexity of the inequivalence problem for regular expressions with intersection. In *Proc. of ICALP-80*, 234–245.

Glimm, B.; Lutz, C.; Horrocks, I.; and Sattler, U. 2008. Conjunctive query answering for the description logic SHIQ. *J. Artif. Intell. Res. (JAIR)* 31:157–204.

Glimm, B.; Horrocks, I.; and Sattler, U. 2008. Unions of conjunctive queries in SHOQ. In *Proc. of KR-08*, 252–262.

Grädel, E., and Walukiewicz, I. 1999. Guarded fixed point logic. In *Proc. of LICS-99*, 45–54.

Gutiérrez-Basulto, V.; Ibáñez-García, Y.; and Jung, J. C. 2017a. Number restrictions on transitive roles in description logics with nominals. In *Proc. of AAAI-17*.

Gutiérrez-Basulto, V.; Ibáñez-García, Y.; and Jung, J. C. 2017b. On query answering in description logics with number restrictions on transitive roles. In *Proc. of DL-17*.

Horrocks, I.; Sattler, U.; and Tobies, S. 2000. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL* 8(3):239–263.

Kaminski, M., and Smolka, G. 2010. Terminating tableaux for $\mathcal{SOQ}$ with number restrictions on transitive roles. In *Proc. of the 6th IFIP TC*, 213–228.

Kazakov, Y., and Pratt-Hartmann, I. 2009. A note on the complexity of the satisfiability problem for graded modal logics. In *Proc. of LICS-09*, 407–416.

Kazakov, Y.; Sattler, U.; and Zolin, E. 2007. How many legs do I have? non-simple roles in number restrictions revisited. In *Proc. of LPAR-07*, 303–317.

Lysenko, A.; Roznovăţ, I. A.; Saqi, M.; Mazein, A.; Rawlings, C. J.; and Auffray, C. 2016. Representing and querying disease networks using graph databases. *BioData Mining* 9(1):23.

Rector, A. L., and Rogers, J. 2006. Ontological and practical issues in using a description logic to represent medical concept systems: Experience from GALEN. In *In Proc. of RW-06*, 197–231.

Stefanoni, G.; Motik, B.; Krötzsch, M.; and Rudolph, S. 2014. The complexity of answering conjunctive and navigational queries over OWL 2 EL knowledge bases. *J. Artif. Intell. Res.* 51:645–705.

Stevens, R.; Aranguren, M. E.; Wolstencroft, K.; Sattler, U.; Drummond, N.; Horridge, M.; and Rector, A. L. 2007. Using OWL to model biological knowledge. *International Journal of Man-Machine Studies* 65(7):583–594.

Vardi, M. Y. 1998. Reasoning about the past with two-way automata. In *Proc. ICALP-98*, 628–641.

Wolstencroft, K.; Brass, A.; Horrocks, I.; Lord, P.; Sattler, U.; Turi, D.; and Stevens, R. 2005. A little semantic web goes a long way in biology. In *Proc. of ISWC-05*.

APPENDIX

Additional Preliminaries

Homomorphisms. Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be two interpretations. A *homomorphism from $\mathcal{I}_1$ to $\mathcal{I}_2$* is a function $h : \Delta^{\mathcal{I}_1} \to \Delta^{\mathcal{I}_2}$ such that *(i)* $h(a) = a$ for all $a \in \mathsf{N_I}$, *(ii)* if $d \in A^{\mathcal{I}_1}$, then $h(d) \in A^{\mathcal{I}_2}$, for all $A \in \mathsf{N_C}$, and *(iii)* if $(d, e) \in r^{\mathcal{I}_1}$, then $(h(d), h(e)) \in r^{\mathcal{I}_2}$, for all $r \in \mathsf{N_R}$. It is folklore that PRQPs are *preserved under homomorphisms*, that is, if $\mathcal{I}_1 \models \varphi$ and there is a homomorphism from $\mathcal{I}_1$ to $\mathcal{I}_2$, then $\mathcal{I}_2 \models \varphi$.

Semantics of 2ATAs. A *run* of $\mathfrak{A}$ on a labelled tree (T, τ) is a $T \times Q$-labelled tree (T_r, r) such that $r(\varepsilon) = (\varepsilon, q_0)$ and whenever $x \in T_r$, $r(x) = (w, q)$, and $\delta(q, \tau(w)) = \theta$, then there is a set $\mathcal{S} = \{(m_1, q_1), \ldots, (m_n, q_n)\} \subseteq [k] \times Q$ such that $\mathcal{S}$ satisfies θ and for $1 \leq i \leq n$, we have $x \cdot i \in T_r$, $w \cdot m_i$ is defined, and $\tau_r(x \cdot i) = (w \cdot m_i, q_i)$. A run is *accepting* if every infinite path π satisfies the *parity condition*. A *parity condition F over Q* is a finite sequence $G_1, \ldots, G_m$ with $G_1 \subseteq G_2 \subseteq \ldots \subseteq G_m = Q$. An infinite path π satisfies F if there is an even i such that $\mathsf{inf}(\pi) \cap G_i \neq \emptyset$ and $\mathsf{inf}(\pi) \cap G_{i-1} = \emptyset$, where $\mathsf{inf}(\pi) \subseteq Q$ denotes the set of states that occur infinitely often in π. The automaton accepts an input tree if there is an accepting run for it. We use $L(\mathfrak{A})$ to denote the set of trees accepted by $\mathfrak{A}$. The *nonemptiness problem* is to decide, given a 2ATA $\mathfrak{A}$, whether $L(\mathfrak{A})$ is nonempty.

A Proof of Lemma 1

Lemma 1. *For each $\mathcal{I} \models \mathcal{K}$, there is a sub-interpretation $\mathcal{I}'$ of $\mathcal{I}$ with $\mathcal{I}' \models \mathcal{K}$ and width and breadth of $\mathcal{I}'$ are bounded by $O(|\mathcal{A}| + 2^{\mathsf{poly}(|\mathcal{T}|)})$.*

Proof. We show the lemma in two stages, adapting a technique from (Kazakov and Pratt-Hartmann 2009; Gutiérrez-Basulto, Ibáñez-García, and Jung 2017a).

Let $\hat{m}$ be the maximal number appearing in $\mathcal{T}$.

Stage 1 (Bounded breadth). As it is standard to achieve bounded breadth for non-transitive roles (Glimm et al. 2008), we only deal with transitive roles here.

An element e is an *strict r-successor of d* if $(d, e) \in r^{\mathcal{I}}$, but $e \notin Q_{\mathcal{I},r}(d)$. Let $W_r(d)$ be the set of strict r-successors of d and $W_r(d, C) \subseteq W_r(d)$ be the set of all strict r-successors of d satisfying C. Then, fix a subset $W'_r(d) \subseteq W_r(d)$ by adding, for each $C \in \mathsf{cl}(\mathcal{T})$, $\min(\hat{m}, |W_r(d, C)|)$ elements from $W_r(d, C)$.

Assume without loss of generality that $W'_r(d_1) = W'_r(d_2)$ if $d_1 \in Q_{\mathcal{I},r}(d_2)$, and define relations S^1_r, S^2_r, and S^3_r, for each $r \in \mathsf{Rol}_t(\mathcal{K})$, as follows:

$$S^1_r = \{(d, d') \in r^{\mathcal{I}} \mid d' \in Q_{\mathcal{I},r}(d)\};$$
$$S^2_r = \{(d, d') \in r^{\mathcal{I}} \mid r(d, d') \in \mathcal{A}\};$$
$$S^3_r = \{(d, d') \in r^{\mathcal{I}} \mid d' \in W'_r(d)\}.$$

Intuitively, S^1_r is the restriction of $r^{\mathcal{I}}$ to the clusters, S^2_r takes care of the ABox, and S^3_r keeps a sufficient set of successors to witness all at-least restrictions.

Finally, obtain $\mathcal{I}'$ from $\mathcal{I}$ by taking $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$, $A^{\mathcal{I}'} = A^{\mathcal{I}}$ for all concept names A, $r^{\mathcal{I}'} = r^{\mathcal{I}}$, for all non-transitive roles r, and, for all transitive roles r,

$$r^{\mathcal{I}'} = (S^1_r \cup S^2_r \cup S^3_r)^+.$$

Claim 1. $C^{\mathcal{I}} = C^{\mathcal{I}'}$, for all $C \in \mathsf{cl}(\mathcal{T})$.

Proof of Claim 1. This is shown by induction on the structure of concepts. The only non-trivial case are concepts $C = (\leqslant n\ r.D)$, r transitive. Clearly, $d \in C^{\mathcal{I}}$ implies $d \in C^{\mathcal{I}'}$ since $r^{\mathcal{I}'} \subseteq r^{\mathcal{I}}$. The converse is a direct consequence of the definition of $W'_r(d)$ and S^3_r. In particular, only r-successors that "cannot be seen" by at-most restrictions (due to the choice of $\hat{m}$) are removed.

From Claim 1, we conclude that $\mathcal{I}' \models \mathcal{T}$; by Claim 1 and the definition of $r^{\mathcal{I}'}$, particularly S^2_r, we also have $\mathcal{I}' \models \mathcal{A}$, thus $\mathcal{I} \models \mathcal{K}$. Since $r^{\mathcal{I}'} \subseteq r^{\mathcal{I}}$ and $A^{\mathcal{I}} = A^{\mathcal{I}'}$, for all $A \in \mathsf{N_C}$, the identity is an homomorphism from $\mathcal{I}'$ to $\mathcal{I}$. Finally note that, by construction, the breadth of $\mathcal{I}'$ is at most $|\mathcal{A}| + |\mathsf{cl}(\mathcal{T})| \cdot \hat{m}$ and thus $O(|\mathcal{A}| + 2^{\mathsf{poly}(|\mathcal{T}|)})$.

Stage 2 (Bounded Width). For every transitive role r, and every $d \in \Delta^{\mathcal{I}}$, fix a set $W_r(d) \subseteq Q_{\mathcal{I},r}(d)$ as follows. For each $C \in \mathsf{cl}(\mathcal{T})$, $W_r(d)$ contains the set $Q_{\mathcal{I},r}(d) \cap C^{\mathcal{I}}$ if this set has size at most $\hat{m}$, and otherwise a subset thereof having size $\hat{m}$. Without loss of generality, we assume that $W_r(d) = W_r(e)$ for all $e \in Q_{\mathcal{I},r}(d)$. Now, define a set Δ_r, for each transitive r, by taking

$$\Delta_r = \mathsf{ind}(\mathcal{A}) \cup \bigcup_{d \in \Delta^{\mathcal{I}}} W_r(d),$$

and define an interpretation $\mathcal{I}' = (\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'})$ by setting $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$, $A^{\mathcal{I}'} = A^{\mathcal{I}}$, for all $A \in \mathsf{N_C}$, $r^{\mathcal{I}'} = r^{\mathcal{I}}$, for all non-transitive roles r, and

$$r^{\mathcal{I}'} = r^{\mathcal{I}} \cap (\Delta^{\mathcal{I}} \times \Delta_r), \text{ for all transitive roles } r.$$

It is not hard to verify that $r^{\mathcal{I}'}$ is indeed transitive.

Claim 2. $C^{\mathcal{I}} = C^{\mathcal{I}'}$, for all $C \in \mathsf{cl}(\mathcal{T})$.

Proof of Claim 2. This is again shown by induction on the structure of concepts. The only non-trivial case are concepts $C = (\leqslant n\ r.D)$, r transitive. Clearly, $d \in C^{\mathcal{I}}$ implies $d \in C^{\mathcal{I}'}$ since $r^{\mathcal{I}'} \subseteq r^{\mathcal{I}}$. The converse is a direct consequence of the definition of $W_r(d)$, in particular the choice of $\hat{m}$, and the definition of $r^{\mathcal{I}'}$. In particular, we remove only r-successors that cannot contribute to at-least restrictions.

Based on Claim 2, it is easy to see that $\mathcal{I}' \models \mathcal{T}$ and $\mathcal{I}' \models \mathcal{A}$. Moreover, the identity is a homomorphism from $\mathcal{I}'$ to $\mathcal{I}$. Finally, by definition of $W_r(d)$, particularly the choice of $\hat{m}$, it should be clear that the width of $\mathcal{I}'$ is bounded by $|\mathcal{A}| + 2^{\mathsf{poly}(|\mathcal{T}|)}$. ☐

B Properties of $\mathsf{Wit}_{\mathcal{I},r}(d)$

We next verify two properties of the witness set $\mathsf{Wit}_{\mathcal{I},r}(d)$, which are needed later on. Throughout the following Lemmas, we denote with $W^{\leadsto}_{\mathcal{I},r}(d)$ the set $\{e \mid d \leadsto^*_{\mathcal{I},r} e\}$.

Lemma 2. *For every $d \in \Delta^{\mathcal{I}}$ and transitive r, we have $|\mathsf{Wit}_{\mathcal{I},r}(d)| \leq |\mathcal{A}| \cdot 2^{p(|\mathcal{T}|)}$, for some polynomial p.*

Proof. We construct a tree T labeled with elements from $\Delta^{\mathcal{I}}$. We start with the single node tree d. Then, we exhaustively performing the following operation:

$(*)$ Choose a leaf labeled with e and add, for all $f \in \Delta^{\mathcal{I}} \setminus T$ with $e \rightsquigarrow_{\mathcal{I},r} f$, f as a successor of e in T.

By definition of $\rightsquigarrow_{\mathcal{I},r}$ and $(*)$, the obtained graph is indeed a tree which additionally satisfies $W^{\rightsquigarrow}_{\mathcal{I},r}(d) \subseteq T$. Now, consider the labelling $\ell : T \to 2^{\mathsf{cl}(\mathcal{T})}$ given by

$$\ell(e) = \{C \mid e \in (\leqslant n\, r.C)^{\mathcal{I}}, (\leqslant n\, r.C) \in \mathsf{cl}(\mathcal{T})\}.$$

Let f be a successor of e in T. By construction of T, this implies

- $\ell(e) \subseteq \ell(f)$ if f is a leaf in T;
- $\ell(e) \subsetneq \ell(f)$ if f is not a leaf in T.

Thus, the depth of T is bounded by $|\mathcal{T}|$. Since, for any e, there are at most exponentially (in $\mathcal{T}$) many f such that $e \rightsquigarrow_{\mathcal{I},r} f$, we know that the outdegree of (T, E) is bounded exponentially in $\mathcal{T}$. Overall, we get that the size of T, and thus of the set $W^{\rightsquigarrow}_{\mathcal{I},r}(d)$ is bounded by an exponential in $\mathcal{T}$. Note next that, by Lemma 1, for every $f \in W^{\rightsquigarrow}_{\mathcal{I},r}(d)$, we have $Q_{\mathcal{I},r}(d) \subseteq \mathsf{ind}(\mathcal{A}) \cup X_d$, for some set X_d of size bounded by $2^{p(|\mathcal{T}|)}$, p a polynomial. As $\mathsf{Wit}_{\mathcal{I},r}(d) = \bigcup_{e \in W^{\rightsquigarrow}_{\mathcal{I},r}(d)} Q_{\mathcal{I},r}(e)$, this implies the statement in the lemma. □

Lemma 7. *Let $d \in \Delta^{\mathcal{I}}$ and r transitive. Then for all $e \in \mathsf{Wit}_{\mathcal{I},r}(d)$, we have that $\mathsf{Wit}_{\mathcal{I},r}(e) \subseteq \mathsf{Wit}_{\mathcal{I},r}(d)$.*

Proof. Let $e \in \mathsf{Wit}_{\mathcal{I},r}(d)$. By definition of $\mathsf{Wit}_{\mathcal{I},r}$, it suffices to show that $W^{\rightsquigarrow}_{\mathcal{I},r}(e) \subseteq \mathsf{Wit}_{\mathcal{I},r}(d)$. To this end, suppose $f \in W^{\rightsquigarrow}_{\mathcal{I},r}(e)$. By definition of $W^{\rightsquigarrow}_{\mathcal{I},r}$, there is a sequence $e_1 \rightsquigarrow_{\mathcal{I},r} \cdots \rightsquigarrow_{\mathcal{I},r} e_n$ with $e = e_1$ and $f = e_n$ (possibly $n = 1$). As $e \in \mathsf{Wit}_{\mathcal{I},r}(d)$, we have either *(i)* $e \in W^{\rightsquigarrow}_{\mathcal{I},r}(d)$ or *(ii)* there is some $e' \in W^{\rightsquigarrow}_{\mathcal{I},r}(d)$ such that $e \in Q_{\mathcal{I},r}(e')$. We distinguish cases.

(i) $e \in W^{\rightsquigarrow}_{\mathcal{I},r}(d)$ implies that there is a sequence $d_1 \rightsquigarrow_{\mathcal{I},r} \cdots \rightsquigarrow_{\mathcal{I},r} d_m$ with $d_1 = d$ and $d_m = e$. Thus, there is a sequence $d_1 \rightsquigarrow_{\mathcal{I},r} \cdots \rightsquigarrow_{\mathcal{I},r} d_m = e = e_1 \rightsquigarrow_{\mathcal{I},r} \cdots \rightsquigarrow_{\mathcal{I},r} e_n = f$. Hence, $f \in W^{\rightsquigarrow}_{\mathcal{I},r}(d) \subseteq \mathsf{Wit}_{\mathcal{I},r}(d)$.

(ii) Similar to Case *(i)*, there is a sequence $d_1 \rightsquigarrow_{\mathcal{I},r} \cdots \rightsquigarrow_{\mathcal{I},r} d_m$ with $d_1 = d$ and $d_m = e'$. If $e = f$, that is $n = 1$ in the sequence above, we know that $f \in Q_{\mathcal{I},r}(e')$ and thus $f \in \mathsf{Wit}_{\mathcal{I},r}(d)$. Otherwise, observe that we can assume that $|Q_{\mathcal{I},r}(d)| \geq 2$ (otherwise $e' = e$ and we are in Case *(i)*). Thus, we have $e' \in (\sim \ell\, r.C)^{\mathcal{I}}$ iff $e \in (\sim \ell\, r.C)^{\mathcal{I}}$, for all $\sim$, ℓ, and C, and hence also $e' \rightsquigarrow_{\mathcal{I},r} e_2$ implying that $d_1 \rightsquigarrow_{\mathcal{I},r} \cdots \rightsquigarrow_{\mathcal{I},r} d_m = e' \rightsquigarrow_{\mathcal{I},r} e_2 \rightsquigarrow_{\mathcal{I},r} \cdots \rightsquigarrow_{\mathcal{I},r} e_n = f$. Hence, $f \in W^{\rightsquigarrow}_{\mathcal{I},r}(d) \subseteq \mathsf{Wit}_{\mathcal{I},r}(d)$.

□

C Proof of Theorem 1

Before we establish Theorem 1, we prove two auxiliary lemmas, which establish how to address in a unique way r-successors in canonical decompositions. For the first auxiliary lemma, observe that as a consequence of Definition 2, particularly, Condition (C$_3$), for every $d \in \Delta^{\mathcal{J}}$, $r \in \mathsf{N}^t_{\mathsf{R}}$, there is a unique node $w \in T$ with $\mathfrak{r}(w) = r$ and $d \in F_r(w)$. We denote this node with $w_{d,r}$.

Lemma 8. *Let $r \in \mathsf{N}^t_{\mathsf{R}}$. For every $u \in T$ with $\mathfrak{r}(u) = r$ and $(d, e) \in r^{\mathfrak{I}(u)}$, exactly one of the following holds:*

- *$w_{d,r} = w_{e,r}$ and $(d, e) \in r^{\mathfrak{I}(w_{d,r})}$;*
- *$w_{e,r}$ is a successor of $w_{d,r}$, $(d, e) \in r^{\mathfrak{I}(w_{e,r})}$ and d belongs to an r-root cluster in $w_{e,r}$;*
- *$w_{e,r}$ is an ancestor of $w_{d,r}$ and $(d, e) \in r^{\mathfrak{I}(w_{d,r})}$.*

Proof. Since $d, e \in \Delta_u$, we know that $w_{d,r}$ and $w_{e,r}$ are either equal to u or ancestors of u. We distinguish three cases:

- If $w_{d,r} = w_{e,r}$, then, by Definition 1, $d, e \in \Delta_{v'}$ for every v' on the path from $w_{d,r}$ to u, and $(d, e) \in r^{\mathfrak{I}(v')}$ for every such v'. Therefore, $(d, e) \in r^{\mathfrak{I}(w_{d,r})}$.
- If $w_{d,r}$ is an ancestor of $w_{e,r}$, then we know by the same reasoning as in the previous point that $(d, e) \in r^{\mathfrak{I}(w_{e,r})}$. Let $w' = w_{e,r} \cdot -1$ (the predecessor of $w_{e,r}$). Since $\mathfrak{T}$ is a canonical decomposition, either (C$_2$) or (C$_3$) applies to w' and $w_{e,r}$. Assume first that $w' = w_{d,r}$.
  - In case of (C$_2$), since $d \in \Delta_{w_{d,r}} \cap \Delta_{w_{e,r}}$ we know that there is a r-root cluster $\mathbf{a} \subseteq \Delta_{w_{e,r}}$ such that $d \in \mathbf{a}$.
  - In case of (C$_3$), let $\mathbf{a} \subseteq F_r(w')$ be the cluster witnessing this. By Item *(b)*, $\mathbf{a}$ is an r-root cluster in $\mathfrak{I}(w_{e,r})$. By definition, we know $e \in F_r(w_{e,r})$ and thus $e \in F(w_{e,r})$. From this and Item *(d)*, we obtain that $d \in \mathbf{a} \cup F(w_{e,r})$, and since $d \notin F(w_{e,r})$ we know $d \in \mathbf{a}$.

  Thus, in both cases, we are in the second case of the lemma. Assume now that $w' \neq w_{d,r}$. We show that it leads to a contradiction in both cases:
  - In case of (C$_2$), since $d \in \Delta_{\mathfrak{I}(w_{e,r})} \cap \Delta_{\mathfrak{I}(w')}$ we know $d \in F(w')$. On the other hand, $d \in F_r(w_{d,r})$ implies that either $d \in F(w_{d,r})$ or $w_{d,r}$ has a predecessor w'' such that $d \in F(w'')$. This is a contradiction since $w' \neq w''$ since $w_{d,r}$ is an ancestor of $w_{e,r}$.
  - In case of (C$_3$), let $\mathbf{a}$ be the r-cluster witnessing this. By definition, $e \in F_r(w_{e,r})$, implies $e \in F(w_{e,r})$. Since $w_{d,r} \neq w'$ but $w_{d,r}$ is an ancestor of $w_{e,r}$, we know that $w' \neq \varepsilon$ and $\mathfrak{r}(w') = r$. From Item *(d)* we obtain that $d \in \mathbf{a} \cup F(w_{e,r})$, and since $d \notin F(w_{e,r})$, we know $d \in \mathbf{a}$. By Item *(a)*, we know that $\mathbf{a} \subseteq F_r(w')$, but then $w' = w_{d,r}$, contradiction.
- If $w_{e,r}$ is an ancestor of $w_{d,r}$, then we know by the reasoning in the first point that $(d, e) \in r^{\mathfrak{I}(w_{d,r})}$; thus, we are in the last case of the lemma.

□

The second auxiliary lemma now provides a way to address r-successors in canonical tree decompositions. For this purpose, we introduce the notion of r-paths. Let $(T, \mathfrak{I}, \mathfrak{r})$ be a canonical decomposition of an interpretation $\mathcal{I}$. An *r-path from d to e in $(T, \mathfrak{I}, \mathfrak{r})$* is a sequence $d_0, w_0, d_1, \ldots, w_{n-1}, d_n$ such that $d = d_0$, $e = d_n$, and $(d_i, d_{i+1}) \in r^{\mathfrak{I}(w_i)}$, for all $0 \leq i < n$. It is *downward* if, for all $0 < i < n$, w_i is a successor of w_{i-1} and d_i is contained in an r-root cluster of w_i.

We then have:

Lemma 9. *Let $(T, \mathfrak{I}, \mathfrak{r})$ be a canonical decomposition of an interpretation $\mathcal{I}$. We have that $(d, e) \in r^{\mathcal{I}}$ iff one of the following is true:*

- *r is non-transitive and $(d, e) \in r^{\mathfrak{I}(\varepsilon)}$ or $(d, e) \in r^{\mathfrak{I}(v)}$ for some successor v of the unique w where d is fresh;*
- *r is transitive and there is an r-path $d_0, w_0, \ldots, d_n$ from d to e in $(T, \mathfrak{I}, \mathfrak{r})$ such that one of the following holds:*
  - ***A*** *$d_0 \in F_r(w_0) \cup F_r(w_0 \cdot -1)$, $d_1 \in F_r(w_0)$, and $d_0, \ldots, d_n$ is downward, or*
  - ***B*** *$d_0 \in F_r(w_0)$, $d_1 \notin F_r(w_0)$, and if $n > 1$, then $d_1, \ldots, d_n$ is downward and $w_1 \cdot -1$ is an ancestor of w_0 with $d_1 \in F_r(w_1 \cdot -1)$.*

Proof. Let first be $(d, e) \in r^{\mathcal{I}}$ for some non-transitive role r. The direction $(\Leftarrow)$ is immediate. For $(\Rightarrow)$, assume that $(d, e) \notin r^{\mathfrak{I}(\varepsilon)}$. By Condition ($C_1$), $(d, e) \notin r^{\mathfrak{I}(w)}$, for all $w \neq \varepsilon$ such that $\mathfrak{r}(w) \neq r$. The statement then follows from Condition (C_2).

Let now be r transitive. Again, the direction $(\Leftarrow)$ is trivial. For $(\Rightarrow)$, $(d, e) \in r^{\mathcal{I}}$ implies, by definition of tree decomposition, that there is an r-path $d_0, w_0, \ldots, w_{n-1}, d_n$ from d to e in $(T, \mathfrak{I}, \mathfrak{r})$. We show first that it is without loss of generality to assume that for all $0 \leq j < n - 1$, we have:

(a) $w_{j+1} = w_j$,

(b) w_{j+1} is a successor of w_j, $d_{j+1} \in F_r(w_j)$ and d_{j+1} belongs to an r-root cluster in $\mathfrak{I}(w_{j+1})$

(c) w_{j+1} is an ancestor of w_j and $d_{j+1} \in F_r(w_{j+1})$, or

(d) The predecessor of w_{j+1} is an ancestor of w_j, $d_j \in F_r(w_j)$, and $d_{j+1} \in F_r(w_{j+1})$.

Observe that, by Lemma 8, we can assume that $w_i \in \{w_{d_i,r}, w_{d_{i+1},r}\}$. Moreover, if $w_i = w_{d_i,r}$, then either $w_{d_i,r} = w_{d_{i+1},r}$ or $w_{d_{i+1},r}$ is an ancestor of $w_{d_i,r}$ and $d_{i+1} \in F_r(w_{j+1})$; if $w_i = w_{d_{i+1},r}$, then $w_{d_{i+1},r}$ is a successor of $w_{d_i,r}$ and $d_{i+1} \in F_r(w_i)$. Let now be $0 \leq j < n - 1$. We distinguish four cases:

- If $w_j = w_{d_j,r}$ and $w_{j+1} = w_{d_{j+1},r}$, then Case (a) or Case (c) applies.
- If $w_j = w_{d_{j+1},r}$ and $w_{j+1} = w_{d_{j+1},r}$, then Case (a) applies.
- If $w_j = w_{d_j,r}$ and $w_{j+1} = w_{d_{j+2},r}$, then Case (b) or (d) applies.
- If $w_j = w_{d_{j+1},r}$ and $w_{j+1} = w_{d_{j+2},r}$, then Case (b) applies.

Note then, that in case (a) is satisfied for some j, we can safely drop d_{j+1} and w_j and the remaining sequence is still an r-path, due to Definition 1 (item 2). So from now on, we assume that for all $0 \leq j < n - 1$, one of (b)–(d) is the case.

If Condition (b) applies for all j then, by the second item in Lemma 8, the r-path is downward and it satisfies **A**. Otherwise, we modify the sequence by performing the following operation exhaustively. Let $0 \leq k < n - 1$ be some index satisfying (c), that is, w_{k+1} is an ancestor of w_k, and let k' be minimal such that all i with $k' \leq i < k$ satisfy (b). If $k' = k$, then do nothing, otherwise we distinguish the following cases:

Case 1: $w_{k+1} = w_j$ for some $k' \leq j < k$. We show inductively that then $(d_i, d_{k+1}) \in r^{\mathfrak{I}(w_i)}$, for all $j \leq i \leq k$. For $i = k$ it is clear by assumption. For the inductive step, assume $j \leq i < k$. Clearly, we have $(d_i, d_{i+1}) \in r^{\mathfrak{I}(w_i)}$ and, by the choice of k and the assumption $w_j = w_k$, also $(d_{k+1}, d_{k+2}) \in r^{\mathfrak{I}(w_j)}$. Moreover, by induction, we can assume that $(d_{i+1}, d_{k+1}) \in r^{\mathfrak{I}(w_{i+1})}$. By the definition of tree decomposition (item 4), we know that $d_{k+1} \in \Delta_{w_i}$; and that $r(d_{i+1}, d_{k+1}) \in r^{\mathfrak{I}(w_i)}$. Further, the definition of tree decomposition yields also $(d_i, d_{k+1}) \in r^{\mathfrak{I}(w_i)}$, thus finishing the inductive step.
This implies $(d_j, d_{k+1}) \in r^{\mathfrak{I}(w_j)}$. Since also $(d_{k+1}, d_{k+2}) \in r^{\mathfrak{I}(w_j)}$, we know $(d_j, d_{k+2}) \in r^{\mathfrak{I}(w_j)}$. Thus, dropping the subsequence

$$d_{j+1}, w_{j+1}, \ldots, w_{k+1}$$

yields an r-path satisfying (b)–(d) for all j.

Case 2: w_{k+1} is an ancestor of $w_{k'}$. We can argue as in Case 1 that $(d_{k'}, d_{k+1}) \in r^{\mathfrak{I}(w_{k'})}$. Thus, we can drop the subsequence $d_{k'+1}, \ldots, d_k, w_k$ obtaining an r-path which satisfies (b)–(d), for all j.

We can deal similarly with an index satisfying (d). After performing this step exhaustively, we obtain an r-path $e_0, v_0, \ldots, v_{m-1}, e_m$ from d to e which is downward, and satisfies **A** or **B**, or

($*$) there is some $0 \leq j < m$ such that (c) holds for all $0 \leq i < j$, and if $j < m-1$, then (d) holds for j, and (b) holds for all $j < i < m$.

In case of ($*$), we show how to obtain an r-path satisfying ($*$) with $j = 0$.

Claim. If $j \geq 1$, then $(e_0, e_2) \in r^{\mathfrak{I}(v_0)}$.

Proof of the Claim. We show inductively that $(e_1, e_2) \in r^{\mathfrak{I}(u)}$ for all u on the path between v_1 and v_0. It is obviously true for $u = v_1$.

Let now u be the successor of some u_0 on the path from v_1 to v_0, and assume by induction that u_0 satisfies $(e_1, e_2) \in \Delta_{u_0}$. Suppose that ($C_3$) holds for u. Then $\mathfrak{r}(u) \neq r$. But since $\mathfrak{r}(w_0) = r$, we know that (C_3) holds again for some node between u and v_0. The only possible witness for this is $e = e_2$. However, this leads to a contradiction as well, because $e_2 \notin F(w)$ for any w on the path between u and v_0. Hence, we know that (C_4) holds for u. Let $\mathbf{a}$ be the r-cluster in $\mathfrak{I}(u)$ witnessing this.

- If $e_2 \in \mathbf{a}$, (C_4) (b) implies that $e_2 \in \Delta_u$, since $(e_1, e_2) \in r^{\mathfrak{I}(u_0)}$.
- If $e_2 \notin \mathbf{a}$, then we know by (C_4) that $(e, e_2) \in r^{\mathfrak{I}(u)}$, for some $e \in \mathbf{a}$. Thus, $(e, e_2) \in r^{\mathfrak{I}(u_0)}$. Again, ($C_4$)(b) implies that $e_2 \in \Delta_u$.

By the definition of tree decomposition, we obtain in both cases $(e_1, e_2) \in r^{\mathfrak{I}(u)}$, thus finishing the induction. Since also $(e_0, e_1) \in r^{\mathfrak{I}(v_0)}$, we obtain $(e_0, e_2) \in r^{\mathfrak{I}(w_0)}$. This finishes the proof of the Claim.

It is now easy to verify that dropping e_1, v_1 from the sequence preserves $(*)$, but with j and m decreased by one. By the Claim, we can perform this operation repeatedly until $j = 0$.

We argue that the remaining r-path satisfies either **A** or **B**.

- If $m = 1$, we distinguish cases according to Lemma 8:
  - if $w_{e_0,r} = w_{e_1,r}$, then $e_0, w_{e_1,r}, e_1$ is a downward path from d to e satisfying **A**;
  - if $w_{e_1,r}$ is a successor of $w_{e_0,r}$, then $e_0, w_{e_1,r}, e_1$ is a downward path from d to e satisfying **A**;
  - if $w_{e_1,r}$ is an ancestor of $w_{e_0,r}$, then $e_0, w_{e_0,r}, e_1$ is an r-path from d to e satisfying **B**.

In case $m > 2$, the resulting path satisfies **B** because of $(*)$, in particular, (d) holds for 0 and (b) holds for all $0 < j < m$. □

We restate Theorem 1 and give the missing details from the proof.

Theorem 1. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{SQ}$ KB and φ a PRPQ with $\mathcal{K} \not\models \varphi$. There is a model $\mathcal{J}$ of $\mathcal{K}$ and a canonical tree decomposition $(T, \mathfrak{I}, \mathfrak{r})$ of $\mathcal{J}$ with (i) $\mathcal{J} \not\models \varphi$, (ii) $\mathfrak{I}(\varepsilon) \models \mathcal{A}$, and (iii) width and outdegree of $(T, \mathfrak{I})$ are bounded by $O(|\mathcal{A}| \cdot 2^{p(|\mathcal{T}|)})$, for some polynomial p.*

Proof. Let $\mathcal{J}$ and $(T, \mathfrak{I}, \mathfrak{r})$ be the interpretation and the tree decomposition obtained by the unraveling procedure in the main part.

We first verify that $(T, \mathfrak{I}, \mathfrak{r})$ is indeed a tree decomposition of $\mathcal{J}$. Items 1 and 2 of $(T, \mathfrak{I})$ being a tree decomposition of $\mathcal{J}$ are an immediate consequence of the definition of $\mathcal{J}$ and $\mathfrak{I}$. Item 4 is a consequence of the nature of the rules. In particular, each rule makes sure that the domain elements in world v are either freshly introduced, or appear in the predecessor.

We argue next that $(T, \mathfrak{I}, \mathfrak{r})$ is canonical. Let $v \in T$ be a successor of $w \in T$ and assume that $r = \mathfrak{r}(w)$ and $s = \mathfrak{r}(v)$. A general property of the construction of $\mathcal{J}$ is that Condition (C_1) is satisfied throughout. More precisely, the application of a rule does not change the interpretation of elements that were already present, and it implies (C_1) in the created interpretation. For the remaining conditions, we distinguish cases which rule has been applied to obtain v from w.

- In case of $\mathbf{R_1}$, it is clear from the definition of $\mathbf{R_1}$, that (C_2) is satisfied.
- If $\mathbf{R_2}$ has been applied, it is clear that $\Delta_v \cap \Delta_w$ is the singleton $\{\delta_0\}$ and that $r \neq s$. By the premise of the rule, we know that $\delta_0 \in F(w)$. By definition of $\mathsf{Wit}_{\mathcal{I},r}$, we know that there is an r-root cluster $\mathbf{a}$ in $\mathfrak{I}(v)$ with $\delta_0 \in \mathbf{a}$. Finally, observe that $\mathbf{R_2}$ is applied only once to every $d \in F(w)$, and $r' \neq r$. Thus, $\mathbf{R_2}$ satisfies (C3).
- Suppose $\mathbf{R_3}$ has been applied to some r-cluster $\mathbf{a}$ in $\mathfrak{I}(w)$ with $\mathbf{a} \subseteq F_r(w)$ and a direct r-successor $\hat{e}$ of $\tau(\delta)$ in $\mathcal{I}$, for some $\delta \in \mathbf{a}$ such that $(\delta, \delta') \notin r^{\mathcal{J}}$, for any δ' with $\tau(\delta') = \hat{e}$. We show that $\mathbf{a}$ witnesses (C4).
  By definition of Δ' and (†), $\mathbf{a}$ is an r-root cluster in $\mathfrak{I}(v)$. Items (C_4)(a) and (C_4)(b) are satisfied by assumption. Item (C_4)(c) follows from the definition of Δ_v. For (C_4)(d), assume $(d, e) \in r^(\mathfrak{I}(v))$ and suppose that $e \in F(v)$. By definition of $\mathbf{R_3}$, we know that $(d, e) \in r^{\mathcal{J}}$. Because $e \in F(v)$, the tuple (d, e) has been added to $r^{\mathcal{J}}$ in this step via the application of (†). Thus, we obtain $d \in \mathbf{a} \cup \Delta = \mathbf{a} \cup F(v)$, as required.

We next verify that $(T, \mathfrak{I}, \mathfrak{r})$ and $\mathcal{J}$ satisfy Conditions *(i)*–*(iii)* from the statement.

Condition *(i)* is a consequence of the fact that τ is a homomorphism from $\mathcal{J}$ to $\mathcal{I}$ and that PRPQs are preserved under homomorphisms. Condition *(ii)* is ensured by the initialization phase. For Condition *(iii)*, we start with the bounding the width. We distinguish cases according to which rule was applied.

- for the root ε of T, we know that $|\Delta_\varepsilon|$ is bounded as required by construction and Lemmas 1 and 2.
- If w was created by $\mathbf{R}_1$, then $|\Delta_w| = 2$.
- If w was created by $\mathbf{R}_2$, then $|\Delta_w|$ is bounded as required by Lemma 2.
- If w was created by $\mathbf{R}_3$, let $\mathcal{C}$ denote the set of all concepts $(\leqslant m\, r.D)$ appearing in $\mathcal{T}$. We make the following observations.
  (a) For all $(d, e) \in r^{\mathcal{I}}$, $C \in \mathcal{C}$: if $d \in C^{\mathcal{I}}$, then $e \in C^{\mathcal{I}}$;
  (b) Let $d_1, \ldots, d_n$ be such that $(d_i, d_{i+1}) \in r^{\mathcal{I}}$, for all $1 \leq i < n$ and $n > |\mathcal{T}|$. Then $\mathsf{Wit}_{\mathcal{I},r}(d_i) = \mathsf{Wit}_{\mathcal{I},r}(d_j)$ for some $i \neq j$.

  Point (a) follows from the semantic of $(\leqslant m\, r.D)$ and transitivity. For Point (b) observe that $|\mathcal{C}| < n$, thus there are $i \neq j$ such that $d_i \in C^{\mathcal{I}}$ iff $d_j \in C^{\mathcal{I}}$, for all $C \in \mathcal{C}$. By definition of $\rightsquigarrow_{\mathcal{I},r}$ and $\mathsf{Wit}_{\mathcal{I},r}$, we also have $\mathsf{Wit}_{\mathcal{I},r}(d_i) = \mathsf{Wit}_{\mathcal{I},r}(d_j)$.
  Now consider some branch of applications of $\mathbf{R}_3$. Each application adds (copies of) elements which are new witnesses, that is, they are in $\mathsf{Wit}_{\mathcal{I},r}(e) \setminus \mathsf{Wit}_{\mathcal{I},r}(d)$, for some $(d, e) \in r^{\mathcal{I}}$. By Point (b), along such a branch, elements are added at most $|\mathcal{T}|$ times. Each time, at most $|\mathsf{Wit}_{\mathcal{I},r}(d)|$ elements are added. Overall, the size is bounded by $|\mathcal{T}| \cdot (|\mathcal{A}| \cdot 2^{p(|\mathcal{T}|)}) = O(|\mathcal{A}| \cdot 2^{p(|\mathcal{T}|)})$.

Finally, the outdegree is bounded by $k_1 \cdot k_2 \cdot k_3$, where k_1 is the number of elements in a bag, k_2 is the number of role names, and k_3 is the maximal outdegree in $\mathcal{I}$. We have seen bounds for k_1 and k_2. So it remains to note that the outdegree in $\mathcal{I}$ is bounded by $|\mathcal{A}| + 2^{\mathsf{poly}(|\mathcal{T}|)}$, by Lemma 1.

It remains to prove that $\mathcal{J} \models \mathcal{K}$, which is a consequence of the following claim.

Claim. For all $\delta \in \Delta^{\mathcal{J}}$ and all $C \in \mathsf{cl}(\mathcal{T})$, we have

$$\delta \in C^{\mathcal{J}} \text{ iff } \tau(\delta) \in C^{\mathcal{I}}.$$

Proof of the Claim. The proof is by induction on the structure of concepts. The case $C = A$ for $A \in \mathsf{N_C}$ follows from τ being a homomorphism and rules $\mathbf{R}_1$–$\mathbf{R}_3$. The Boolean cases $C = \neg D$ and $C = C_1 \sqcap C_2$ are consequences of the induction hypothesis. It thus remains to consider concepts of the form $C = (\sim n\, r.D)$. If $r \in \mathsf{N}_\mathsf{R}^{nt}$, the claim is a straightforward consequence of the induction hypothesis and construction rule $\mathbf{R}_1$. Now, assume that $r \in \mathsf{N}_\mathsf{R}^{t}$. It suffices to show that:

(a) If $\tau(\delta) \notin (\leqslant n\, r.D)^{\mathcal{I}}$, then $\delta \notin (\leqslant n\, r.D)^{\mathcal{J}}$, and

(b) if $\tau(\delta) \in (\leqslant n\, r.D)^{\mathcal{I}}$, then $\delta \in (\leqslant n\, r.D)^{\mathcal{J}}$.

For Point (a) Let now be $\tau(\delta) = d$ and $d \notin (\leqslant n\, r.D)^{\mathcal{I}}$, that is $d \in (\geqslant (n+1)\, r.D)^{\mathcal{I}}$. It suffices to show that δ has $n+1$ r-successors satisfying D. Let $d' \in \Delta^{\mathcal{I}}$ be any domain element such that $(d, d') \in r^{\mathcal{I}}$ and $d' \in D^{\mathcal{I}}$. Thus, either $d' \in Q_{\mathcal{I},r}(d)$ or there is a sequence $d_0, \dots, d_m$ with $d_0 = d$, $d_m = d'$, and d_{i+1} is a direct r-successor of d_i in $\mathcal{I}$, for all $0 \leq i < m$.

In the first case, by construction, there is $\delta' \in Q_{\mathcal{J},r}(\delta)$ with $\tau(\delta') = d'$. By induction hypothesis, $\delta' \in D^{\mathcal{J}}$.

In the second case, we show that for every such sequence, every $\delta \in \Delta^{\mathcal{J}}$ with $\tau(\delta) = d_0$ and every $0 \leq i \leq m$ there is an r-path from δ to some δ_i with $\tau(\delta_i) = d_i$. The base case $i = 0$ is immediate. So suppose there is an r-path $\pi = \delta_0, w_0, \dots, w_{i-1}, \delta_i$ from δ_0 to δ_i with $\tau(\delta_i) = d_i$, for $i > 0$. Let $w \in T$ be such that $w = w_{d_i}$, and $\mathbf{a} \subseteq F_r(w)$ with the r-cluster in $\mathfrak{I}(w)$ such that $d_i \in \mathbf{a}$. We distinguish two cases:

- If there is some $\delta' \in \Delta^{\mathcal{J}}$, such that $(\delta, \delta') \in r^{\mathcal{J}}$, for some with $\tau(\delta') = d_{i+1}$. Then, there is some r-path $\delta, v_0, \dots, v_k, \delta'$ from δ to δ' in $(T, \mathfrak{I}, \mathfrak{r})$. Then $\pi, v_0, \dots, v_k, \delta'$ is the required r-path.
- If $(\delta, \delta') \notin r^{\mathcal{J}}$, for all δ' with $\tau(\delta') = d_{i+1}$, then $\mathbf{R}_3$ applies to w, $\mathbf{a}$, and d_{i+1}. In particular, it adds a successor v of w to T and adds a domain element $\delta' = (d_{i+1})_v \in \Delta$ to $\mathcal{J}$ such that $(\delta_i, \delta') \in r^{\mathcal{J}}$. By construction, $(\delta, \delta') \in r^{\mathfrak{I}(v)}$ and π, v, δ' is the required r-path.

Thus, we can conclude that there is an r-path from δ to some δ' with $\tau(\delta') = d'$. By induction, we know that $\delta' \in D^{\mathcal{J}}$. Since distinct d' with $(d, d') \in r^{\mathcal{I}}$ and $d' \in D^{\mathcal{I}}$ yield distinct δ', this finishes the proof of (a).

For Point (b) Assume $\tau(\delta) \in (\leqslant n\, r.D)^{\mathcal{I}}$ with $\tau(\delta) = d$. It clearly suffices to show that for every $e \in \mathsf{Wit}_{\mathcal{I},r}(d)$, there is at most one $\delta' \in \Delta^{\mathcal{J}}$ with $(\delta, \delta') \in r^{\mathcal{J}}$. To do so, let w be the (unique) world where δ is r-fresh in $(T, \mathfrak{I}, \mathfrak{r})$. We show first that

(x) for every $e \in \mathsf{Wit}_{\mathcal{I},r}(d)$, there is precisely one $\delta' \in \Delta_w$ with $(\delta, \delta') \in r^{\mathfrak{I}(w)}$ and $\tau(\delta') = e$.

(xx) for every $\delta' \in \Delta^{\mathcal{J}}$ such that $(\delta, \delta') \in r^{\mathcal{J}}$ and $\tau(\delta') \in \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta))$, we have $\delta' \in \Delta_w$

For showing *(x)*, observe that either $w = \varepsilon$ or δ was added either by an application of $\mathbf{R}_2$ or $\mathbf{R}_3$. We distinguish cases:

- Suppose first $w = \varepsilon$. If $\delta = a \in \mathsf{ind}(\mathcal{A})$ then *(x)* is clear due to the initialization of $\mathcal{J}$ and $(T, \mathfrak{I}, \mathfrak{r})$. If $\delta = d_r \in \Delta^r$, we have that $d \in \mathsf{Wit}_{\mathcal{I},r}(a)$ for some individual a, and by Lemma 7 thus $\mathsf{Wit}_{\mathcal{I},r}(d) \subseteq \mathsf{Wit}_{\mathcal{I},r}(a)$. Together with the initialization of $\mathcal{J}$ and of $(T, \mathfrak{I}, \mathfrak{r})$, this yields that Δ_ε contains exactly one element δ_e such that $(\delta, \delta_e) \in r^{\mathfrak{I}(\varepsilon)}$ and $\tau(\delta_e) = e$, for each $e \in \mathsf{Wit}_{\mathcal{I},r}(d)$.
- If w was created by $\mathbf{R}_2$, then there is some $\hat{\delta} \in \Delta_w \cap \Delta_{w'}$ such that for every $\delta' \in \Delta_w$, with $\hat{\delta} \neq \delta'$, $\tau(\delta') \in \mathsf{Wit}_{\mathcal{I},r}(\tau(\hat{\delta}))$. The claim *(x)* follows now by Lemma 7 and the definition of $\mathbf{R}_2$.
- If w was created by $\mathbf{R}_3$, let $\mathbf{a} \subseteq F_r(w \cdot -1)$ be the r-cluster in Δ_w that witnesses this. Then there is some $\hat{\delta} \in \mathbf{a}$ and $\delta' \in \Delta_w$ such that $\tau(\delta')$ is a direct successor of $\tau(\hat{\delta})$. Further, by the choice of w, we have $\delta \in F(w)$. Then, by the definition of $\mathbf{R_3}$, $\tau(\delta) \in \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta')) \setminus \mathsf{Wit}_{\mathcal{I},r}(\tau(\hat{\delta}))$. By Lemma 7, the definition of $\mathbf{R}_3$ ensures that there is exactly one fresh element in Δ_w for every $e \in \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta)) \setminus \mathsf{Wit}_{\mathcal{I},r}(\tau(\hat{\delta}))$. It remains to show that there is exactly one element in Δ_w for every $e \in \mathsf{Wit}_{\mathcal{I},r}(\hat{\delta}) \cap \mathsf{Wit}_{\mathcal{I},r}(\delta')$. Indeed, we can (inductively) assume that *(x)* holds for $w' = w \cdot -1$ and $\hat{\delta}$, that is, for every such e, there is some $d'' \in \Delta_{w'}$ such that $\tau(\delta'') = e$. Moreover $(\hat{\delta}, \delta'') \in r^{\mathfrak{I}(w')}$, and by the definition of $\mathbf{R}_3$, we have $\delta'' \in \Delta_w$.

For showing *(xx)*, assume $(\delta, \delta') \in r^{\mathcal{J}}$. By Lemma 9, there is an r-path $\pi = \delta_0, w_0, \dots, w_{k-1}, \delta_k$ from δ to δ', satisfying either $\mathbf{A}$ or $\mathbf{B}$. We use the following auxiliary claims.

Claim 1. *For every v, and every $\delta \in \Delta_w$. If v was created by an application of rule $\mathbf{R}_3$ and $\delta \notin F_r(v)$ then $W \subseteq \Delta_{v \cdot -1}$ where*

$$W = \{\delta' \in \Delta_v \mid (\delta, \delta') \in r^{\mathfrak{I}(v)} \wedge \tau(\delta') \in \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta))\}.$$

Proof. Let $\mathbf{a}$ in $\mathfrak{I}(w)$ be the r-cluster used in the application of $\mathbf{R}_3$, $w = v \cdot -1$, and $\delta' \in W$. We know that v satisfies (C_4) and that $\mathbf{a}$ witnesses this. Thus, $\mathbf{a}$ is an r-root cluster in $\mathfrak{I}(w)$ such that $\mathbf{a} \subseteq F_r(w)$. Since $(\delta, \delta') \in r^{\mathfrak{I}(w)}$, we have that $\delta \in \Delta_v$, which by $\mathbf{R}_3$ means that either $\delta \in \mathbf{a}$, or there is some $\hat{\delta} \in \mathbf{a}$ such that $(\hat{\delta}, \delta) \in r^{\mathfrak{I}(w)}$, and thus $\delta \notin \mathbf{a} \cup F(v)$.

If $\delta \notin \mathbf{a} \cup F(v)$, then $(\delta, \delta') \in r^{\mathfrak{I}(v)}$ implies that $\delta' \notin F(v)$, by condition (C_4)(d), that is $\delta' \in \Delta_w$. Now, assume $\delta \in \mathbf{a}$. In that case, since $\tau(\delta') \in \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta))$ we know that δ' is not one of the fresh elements of the form f_v added by $\mathbf{R}_3$. Therefore, $\delta' \notin F(v)$ and $\delta' \in \Delta_w$. □

Claim 2. *Let $\pi = \delta_0, w_0, \dots, w_{k-1}, \delta_k$ be a downward r-path with $\tau(\delta_k) \in \mathsf{Wit}_{\mathcal{I}_r}(\tau(\delta_0))$. Then, for every $0 \leq i \leq k-1$, we have $\delta' \in \Delta_{w_i}$, and $(\delta_i, \delta') \in r^{\mathfrak{I}(w_i)}$.*

Proof. We show this using an inductive argument. This holds by assumption for $i = k-1$ and the definition of r-path, and since $(\delta_0, \delta_{k-1}) \in r^{\mathcal{J}}$ and $\tau(\delta') \in \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta_0))$ implies $\tau(\delta') \in \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta_{k-1}))$. For the inductive step, assume that this holds for $0 < i+1 \leq k-1$. Then w_{i+1} was created by an application of $\mathbf{R}_3$. Since $(\delta_i, \delta_{i+1}) \in r^{\mathfrak{I}w_i}$, then $w_{i+1} \notin F_r(w_{i+1})$. Then, as $(\delta_0, \delta_{i+1}) \in r^{\mathcal{J}}$ and $\tau(\delta') \in \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta_0))$ we can conclude $\tau(\delta') \in \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta_{k-1}))$. Thus by Claim 1 $\delta' \in \delta_i$. □

We now do a final case distinction according to which case **A** or **B** applies to π.

- Assume that π satisfies **A**. From Claim 2 we can conclude that $\delta' \in \Delta_0$. By **A**, we have that either $w_0 = w$, and then $\delta' \in \Delta_w$ as required; or $w = w \cdot -1$, and then it must be the case that w_0 was created by an application or rule $\mathbf{R}_3$. Thus the statement follows from Claim 2.
- Assume that π satisfies **B**. The statement clearly holds if $k = 0$. For $k > 1$, using Claim 2 we can conclude $\delta' \in \Delta_{w_1}$. We know from **B** that $\delta_1 \in F_r(w_1 \cdot -1)$ and thus $\mathfrak{r}(w_1 \cdot -1) = r$. Since $(\delta_1, \delta_2) \in r^{\mathfrak{I}(w_1)}$, by ($\mathrm{C}_1$), we also have that $\mathfrak{r}(w_1) = r$, which means that w_1 was created by rule $\mathbf{R}_3$ and, by Claim 1, we get $\delta' \in w_1 \cdot -1$.
  By **B**, we know that $w_1 \cdot -1$ is an ancestor of w_0. Let $v_0, \ldots, v_n = w_0$ be the path from $w_1 \cdot -1 = v_0 \cdot -1$ to w_0, for $0 \leq n$. From the construction of $\mathcal{J}$, we know that every node v_i was added by an application of either $\mathbf{R}_2$ or $\mathbf{R}_3$. We claim that every v_i was created by $\mathbf{R}_3$. Indeed, we have by definition of tree decomposition (item 4) that $\delta_1 \in \Delta_{v_i}$ for every $0 \leq i \leq n$, since $\delta_1 \in \Delta_{w_0}$. For $n = 0$, the claim follows since $\Delta_{v_0} \cap \Delta_{v_0 \cdot -1} = \delta_1$, $\mathfrak{r}(v_0) = r$, and $(\delta_1, \delta') \in r^{\mathfrak{I}(v_0 \cdot -1)}$. This means that v_0 was not created by $\mathbf{R}_2$. Now assume v_n was created by $\mathbf{R}_3$ for $0 \leq n$. The claim follows for v_{n+1} since $\Delta_{v_n} \cap \Delta_{v_{n+1}} = \delta_1$ and $\delta_1 \notin F(v_n)$ because $\delta_1 \in \Delta_{v_0 \cdot -1}$. Hence, v_{n+1} was not created by $\mathbf{R}_2$.
  Finally, we show that $\delta' \in \Delta_{v_i}$ for every $0 \leq i \leq n$. From $\delta_1 \in \Delta_{v_i \cdot -1} \cap \Delta_{v_i}$ and $(\delta, \delta') \in r^{\mathfrak{I}(v_i \cdot -1)}$ we get $\delta' \in \Delta_{v_1}$. Further, as v_i was introduced via an application of $\mathbf{R}_3$ to some r-cluster $\mathbf{a}$ in $\mathfrak{I}(w)$, we also have that v_i satisfies (C_4) and that $\mathbf{a}$ witnesses this. Then, $\delta_1 \notin F(v)$ implies that either $\delta_1 \in \mathbf{a}$, or there is some $\hat{\delta} \in \mathbf{a}$ such that $(\hat{\delta}, \delta_1) \in r^{\mathfrak{I}(v_i \cdot -1)}$. In either case, by ($\mathrm{C}_4$)(c), we can conclude that $d' \in \Delta_{v_i}$. Therefore, the previous inductive argument together with the fact that $d_1 \in \Delta_{v_0 \cdot -1}$ and $(\delta_1, \delta') \in r^{\mathfrak{I}v_0 \cdot -1}$ imply that $\delta' \in \Delta_{v_i}$ for every i. This in particular implies $\delta' \in \Delta_{w_0}$.

□

D Proof of Theorem 2

Theorem 2. *PRPQ entailment over $\mathcal{SQ}$-knowledge bases is* 2EXPTIME*-complete.*

Proof. The lower bound is inherited from 2EXPTIME-hardness of positive existential query entailment in $\mathcal{ALC}$.

For the upper bound, we first show correctness of the given procedure, that is, we show that $\mathcal{K} \models \varphi$ iff $L(\mathfrak{A}_{\mathsf{can}} \wedge \mathfrak{A}_{\mathcal{K}} \wedge \neg\mathfrak{A}_{\varphi}) \neq \emptyset$.

($\Leftarrow$) Assume that $\mathcal{K} \not\models \varphi$. By Theorem 1, we know that there is a model $\mathcal{J}$ of $\mathcal{K}$ and a canonical tree decomposition $(T, \mathfrak{I}, \mathfrak{r})$ of $\mathcal{J}$ satisfying Conditions *(i)–(iii)*.

The key observation is that, by the size $2K$ of Δ, it is possible to select a mapping $\pi : \Delta^{\mathcal{J}} \to \Delta$ such that for each $w \in T \setminus \{\varepsilon\}$ and each $d \in \Delta_w \setminus \Delta_{w \cdot -1}$, we have $\pi(d) \notin \{\pi(e) \mid e \in \Delta_{w \cdot -1}\}$. Define a Σ-labeled tree (T, τ) by setting, for all $w \in T$, $\mathfrak{I}_w$ to the image of $\mathfrak{I}(w)$ under π and r_w to $\mathfrak{r}(w)$. Clearly, (T, τ) is consistent, and $\mathcal{I}_{(T,\tau)}$ is isomorphic to $\mathcal{J}$. It is not hard to see that $(T, \tau) \in L(\mathfrak{A}_{\mathsf{can}})$. Now, by Lemma 4 and $\mathcal{J} \models \mathcal{K}$, we have $(T, \tau) \in L(\mathfrak{A}_{\mathcal{K}})$, and, by Lemma 6 and $\mathcal{J} \not\models \varphi$, we have $(T, \tau) \notin L(\mathfrak{A}_{\varphi})$. Thus $L(\mathfrak{A}_{\mathsf{can}} \wedge \mathfrak{A}_{\mathcal{K}} \wedge \neg\mathfrak{A}_{\varphi})$ is not empty.

For the direction ($\Rightarrow$), let $(T, \tau) \in L(\mathfrak{A}_{\mathsf{can}} \wedge \mathfrak{A}_{\mathcal{K}} \wedge \neg\mathfrak{A}_{\varphi})$. Since $(T, \tau) \in L(\mathfrak{A}_{\mathsf{can}})$, we know that (T, τ) is consistent, that the represented $(T, \mathfrak{I}, \mathfrak{r})$ is a canonical decomposition of $\mathcal{I}_{(T,\tau)}$. It remains to note that, by Lemmas 4 and 6, we have that $\mathcal{I}_{(T,\tau)} \models \mathcal{K}$ and $\mathcal{I}_{(T,\tau)} \not\models \varphi$, respectively.

The 2EXPTIME-upper bound follows now from the following facts. By Lemma 4 and 6, the construction of the respective automata can be done in (worst case) double exponential time; moreover, the automata have exponentially many states. Since intersection and complement of 2ATAs can be done in polynomial time, we know that $\mathfrak{A}_{\mathsf{can}} \wedge \mathfrak{A}_{\mathcal{K}} \wedge \neg\mathfrak{A}_{\varphi}$ has exponentially many states, and can be constructed in exponential time. It remains to note that emptiness of that automaton can be checked in double exponential time. □

E The Automaton $\mathfrak{A}_{\mathsf{can}}$

We refrain from giving the automaton explicitely, but rather describe its functioning. Let (T, τ) be a Σ-labeled tree. Consistency of (T, τ) can be checked by verifying that:

- r_w is a role name from $\mathcal{K}$ for every $w \neq \emptyset$, and $r_\varepsilon = \bot$, and
- for every $w \in T$, every successor v of w, and any two elements $d, e \in \Delta^{\mathcal{I}_w} \cap \Delta^{\mathcal{I}_v}$, we have that $d \in A^{\mathcal{I}_w}$ iff $e \in A^{\mathcal{I}_v}$, for all concept names A appearing in $\mathcal{K}$, and $(d, e) \in r^{\mathcal{I}_w}$ iff $(d, e) \in r^{\mathcal{I}_v}$, for all role names r appearing in $\mathcal{K}$.

Both can be easily done with a 2ATA.

For verifying that the encoded structure $\mathcal{I}_{(T,\tau)}$ is canonical, we formulate the following variants (C_1')–(C_4') which talk about (T, τ). We call (T, τ) *canonical* iff for every $w \in T$ with $\tau(w) = (\mathcal{I}_w, r)$ and every successor v of w with $\tau(v) = (\mathcal{I}_v, s)$, the following conditions are satisfied:

(C_1') if $(d, e) \in s_1^{\mathcal{I}_v}$, then $s_1 = s$, or $d = e$ and $s_1 \in \mathsf{N}_{\mathsf{R}}^t$;

(C_2') if $s \in \mathsf{N}_{\mathsf{R}}^{nt}$, then $\Delta^{\mathcal{I}_v} = \{d, e\}$, for some $d \in F(w)$, $e \in F(v)$, and $s^{\mathcal{I}_v} = \{(d, e)\}$;

(C_3') if $s \in \mathsf{N}_{\mathsf{R}}^t$ and $r \notin \{\bot, s\}$, then there are $d \in F(w)$ and an r-root cluster $\mathbf{a}$ in $\mathcal{I}_v$ such that $\Delta^{\mathcal{I}_w} \cap \Delta^{\mathcal{I}_v} = \{d\}$ and $d \in \mathbf{a}$; moreover, there is no successor $v' \neq v$ of w satisfying this for d and $r_v' = s$;

(C_4') if $s \in \mathsf{N}_{\mathsf{R}}^t$ and $r \in \{\bot, s\}$, then there is an s-root cluster $\mathbf{a}$ in $\mathcal{I}_v$ with:

(a) $\mathbf{a} \subseteq F_s(w)$;
(b) $\mathbf{a}$ is an s-cluster in $\mathcal{I}_w$;
(c) for all $d \in \mathbf{a}$ and $(d,e) \in s^{\mathcal{I}_w}$, we have $e \in \Delta^{\mathcal{I}_v}$;
(d) for all $(d,e) \in s^{\mathcal{I}_v}$, $d \in \mathbf{a} \cup F(v)$ or $e \notin F(v)$.

It is not difficult to verify that (T,τ) is canonical iff the represented extended tree decomposition $(T,\mathfrak{I},\mathfrak{r})$ is canonical. Moreover, Conditions (C_1')–(C_4') can be implemented in a 2ATA in a straightforward way.

F Proof of Lemma 3

Note that Lemma 3 from the main part is just the reformulation of Lemma 9 adapted to the encoding. We refer the reader to Appendix C for a proof of that Lemma.

G Knowledge Base Automaton $\mathfrak{A}_\mathcal{K}$

Lemma 10. *There is a 2ATA $\mathfrak{A}_\mathcal{K}$ such that for every $(T,\tau) \in L(\mathfrak{A}_{\mathsf{can}})$, we have that $(T,\tau) \in L(\mathfrak{A}_\mathcal{K})$ iff $\mathcal{I}_{T,\tau} \models \mathcal{K}$. It can be constructed in time double exponential in $|\mathcal{K}|$, and has exponentially many states in $|\mathcal{K}|$.*

Proof. $\mathfrak{A}_\mathcal{K}$ is the intersection of two automata $\mathfrak{A}_\mathcal{A}$ and $\mathfrak{A}_\mathcal{T}$ verifying that the input satisfies the ABox and the TBox, respectively. For devising $\mathfrak{A}_\mathcal{A}$, recall that $\mathsf{ind}(\mathcal{A}) \subseteq \Delta$ and we identify, in the translation to (T,τ), $[\varepsilon]_a$ with a, for each $a \in \mathsf{ind}(\mathcal{A})$, to reflect the SNA. Moreover, recall that, by Theorem 1, we can assume that the ABox is actually satisfied in the root. Thus, $\mathfrak{A}_\mathcal{A}$ only needs to check whether $\mathcal{A}$ is satisfied at the root. $\mathfrak{A}_\mathcal{A}$ is a single state automaton $\mathfrak{A}_\mathcal{A} = (\{q_0\},\Sigma,q_0,\delta,F)$ with

$$\delta(q_0,\bullet) = \mathsf{false}$$
$$\delta(q_0,(\mathcal{I},x)) = \text{if } \mathcal{I} \models \mathcal{A}, \text{ then } \mathsf{true} \text{ else } \mathsf{false}$$

The next lemma concentrates on the construction of $\mathfrak{A}_\mathcal{T}$.

Lemma 4 *For every $(T,\tau) \in L(\mathfrak{A}_{\mathsf{can}})$, we have $(T,\tau) \in L(\mathfrak{A}_\mathcal{T})$ iff $\mathcal{I}_{(T,\tau)} \models \mathcal{T}$. It can be constructed in time double exponential in $|\mathcal{K}|$, and has exponentially many states in $|\mathcal{K}|$.*

Proof. For $\mathfrak{A}_\mathcal{T}$, we give the missing transitions and states (see Section 4.1). First, we show how the announced "navigation" works. First, for non-transitive roles, we use the following transitions to navigate the automaton to the unique world where d is fresh, for all states $(\sim n\, r\, D)(d)$:

$$\delta((\sim n\, r\, D)(d),(\mathcal{I},x)) = \mathsf{false} \qquad \text{if } d \notin \Delta^\mathcal{I}$$
$$\delta((\sim n\, r\, D)(d),(\mathcal{I},x)) = \big((0,F_d) \wedge (0,q^*_{(\sim n\, r\, D),d})\big) \vee (-1,(\sim n\, r\, D)(d)) \quad \text{if } d \in \Delta^\mathcal{I}$$
$$\delta(F_d,(\mathcal{I},x)) = \begin{cases} \mathsf{true} & \text{if } x = \varepsilon \\ (-1,\overline{F}_d) & \text{otherwise} \end{cases}$$
$$\delta(\overline{F}_d,(\mathcal{I},x)) = \begin{cases} \mathsf{true} & \text{if } d \notin \Delta^\mathcal{I} \\ \mathsf{false} & \text{otherwise} \end{cases}$$

For transitive roles, we use the following transitions to navigate the automaton to the unique world where d is r-fresh, for all states $(\sim n\, r\, D)(d)$:

$$\delta((\sim n\, r\, D)(d),(\mathcal{I},x)) = \mathsf{false} \qquad \text{if } d \notin \Delta^\mathcal{I}$$
$$\delta((\sim n\, r\, D)(d),(\mathcal{I},x)) = \big((0,F_{r,d}) \wedge (0,q^*_{(\sim n\, r\, D),d})\big) \vee \bigvee_{i\in[k]} (i,(\sim n\, r\, D)(d)) \quad \text{if } d \in \Delta^\mathcal{I},$$

where the transitions for $F_{r,d}$ are given in the main part.

Counting for non-transitive Roles

The automaton implements the strategy suggested by Lemma 3 via the following transitions. We first concentrate on at-least restrictions, so let us fix a state $q^*_{(\geqslant n\ r.D),d}$. Recall that k is the bound on the outdegree, and let $N_{m,k}$ be the set of m-element subsets of $[1,k]$. The automaton then has the following transitions for symbols $(\mathcal{I},s) \in \Sigma$ with $s \neq \bot$, that is, for non-root worlds:

$$\delta(q^*_{(\geqslant n\ r.D),d},(\mathcal{I},s)) = \bigvee_{X\in N_{n,k}} \bigwedge_{i\in X} (i,q_{d,r,D})$$
$$\delta(q_{d,r,D},(\mathcal{I},x)) = \begin{cases} D(e) & \text{if } x = r \text{ and } (d,e) \in r^\mathcal{I} \\ \mathsf{false} & \text{otherwise} \end{cases}$$

For symbols $(\mathcal{I},\bot) \in \Sigma$, we have to additionally take successors in $\mathcal{I}$ into account, which is implemented as follows. Let $S_{\mathcal{I},r}(d)$ denote the set of all e with $(d,e) \in r^\mathcal{I}$. We then define the transition for $\delta(q^*_{(\geqslant n\ r.D),d},(\mathcal{I},\bot))$ as

$$\bigvee_{S\subseteq S_{\mathcal{I},r}(d)} \Big(\bigwedge_{e\in S} (0,D(e)) \wedge \bigvee_{X\in N_{n-|S|,k}} \bigwedge_{i\in X} (i,q_{d,r,D}) \Big)$$

For states corresponding to at-most restrictions, $q^*_{(\leqslant n\ r.D),d}$, we include the complementary transitions, that is, for $(\mathcal{I},s) \in \Sigma$ with $s \neq \bot$:

$$\delta(q^*_{(\leqslant n\ r.D),d},(\mathcal{I},s)) = \bigwedge_{X\in N_{n,k}} \bigvee_{i\in X} (i,\overline{q}_{d,r,D})$$
$$\delta(\overline{q}_{d,r,D},(\mathcal{I},x)) = \begin{cases} (\approx D)(e) & \text{if } x = r \text{ and } (d,e) \in r^\mathcal{I} \\ \mathsf{true} & \text{otherwise} \end{cases}$$

where $\approx D$ denotes the negation normal form of $\neg D$. Moreover, we define the transition for $\delta(q^*_{(\leqslant n\ r.D),d},(\mathcal{I},\bot))$ as

$$\bigwedge_{S\subseteq S_{\mathcal{I},r}(d)} \Big(\bigvee_{e\in S} (0,\approx D(e)) \vee \bigwedge_{X\in N_{n-|S|,k}} \bigvee_{i\in X} (i,\overline{q}_{d,r,D}) \Big)$$

At-most Restrictions (Transitive Roles)

Finally, the following are the transitions for the at-most restrictions (for transitive roles). The strategy there is to try to find $n+1$ r-successors satisfying D and accept if this fails, thus "complementing" the strategy for the at-least restrictions. Let N be the set of all tuples $\mathbf{n} = (n_1,\ldots,n_\ell)$ such that $\sum_i n_i = n+1$. Then, $\delta(q^*_{(\leqslant n\ r.D),d},(\mathcal{I},x))$ is defined as follows:

$$\bigwedge_{\mathbf{n}\in N} \bigwedge_{X\subseteq[1,\ell]} \bigvee_{i\in X} (0,q^{\mathbf{A}}_{(\leqslant n_i\ r.D),a_i}) \wedge \bigwedge_{i\in[1,\ell]\setminus X} (0,q^{\mathbf{B}}_{(\leqslant n_i\ r.D),a_i})$$

$$\delta(q^{\mathbf{A}}_{(\leqslant n\ r.D),d},(\mathcal{I},x)) = (0,\overline{F}_{r,d}) \vee (0,q^{\downarrow}_{(\leqslant n\ r.D),d})$$

$$\delta(q^{\mathbf{B}}_{(\leqslant n\ r.D),d}, (\mathcal{I}, x)) = (0, F_{r,d}) \vee (-1, q^{\uparrow}_{(\leqslant n\ r.D),d})$$

Further, $\delta(q^{\downarrow}_{(\leqslant n\ r.D),d}, (\mathcal{I}, x))$ is defined as (where $\mathbf{m}$ and M are as in Section 4.1):

$$\bigwedge_{\mathbf{m}\in M} (0, p^{\mathsf{loc}'}_{n_0,r,D,d}) \vee \bigvee_{i=1}^{k} (i, p^{\mathsf{succ}}_{(\leqslant n_i\ r.D),d}).$$

For states of the form $p^{\mathsf{loc}'}_{n,r,D,d}$, the transition function is defined as

$$\delta(p^{\mathsf{loc}'}_{n,r,D,d}, (\mathcal{I}, x)) = \bigwedge_{Y\subseteq Q_{\mathcal{I},r}(d), |Y|=n} \bigvee_{e\in Y} \approx D(e).$$

For states of the form $p^{\mathsf{succ}}_{(\leqslant n\ r.D)}$, the transition function on input $\bullet$ is defined as

$$\delta(p_{(\leq n\, r\, D),d}, \bullet) = \begin{cases} \mathsf{true} & \text{if } n > 0 \\ \mathsf{false} & \text{otherwise} \end{cases}$$

On inputs of the form $(\mathcal{I}, x)$, we set $\delta(p^{\mathsf{succ}}_{(\leqslant n\ r.D),d}, (\mathcal{I}, x)) = \mathsf{true}$ if $x \neq r$ or d is not in a root cluster; otherwise, we define

$$\delta(p^{\mathsf{succ}}_{(\leqslant n\ r.D),d}, (\mathcal{I}, x)) = \bigwedge_{\mathbf{n}\in N} \bigvee_{i=1}^{\ell} (0, \overline{q}_{(\leqslant n_i\ r.D),a_i})$$

Finally, we set $\delta(\overline{q}_{(\leqslant n\ r.D),d}, (\mathcal{I}, x)) = \mathsf{true}$ in case $d \notin \Delta^{\mathcal{I}}$, and otherwise

$$\delta(\overline{q}_{(\leqslant n\ r.D),d}, (\mathcal{I}, x))(0, q^{\mathbf{A}}_{(\leqslant n\ r.D),d}) \wedge (0, q^{\mathbf{B}}_{(\leqslant n\ r.D),d}).$$

It remains to define the acceptance condition F. We set $F = G_1, G_2, G_3$ where $G_1 = \emptyset$, G_2 contains all states of the form $q^{*}_{(\sim n\ r.D),d}$ and $p^{\mathsf{loc}}_{n,r,D,d}$ with $n \geq 1$, and $G_3 = Q$. Note that, as mentioned in Section 4.1, the parity condition enforces that states $q^{\downarrow}_{(\geq n\, r\, D),d}$ with $n \geq 1$ are not suspended forever, that is, eventualities are finally satisfied. □

Finally, it is not hard to see that the number of states of the automaton $\mathfrak{A}_{\mathcal{K}}$ (in particular that of $\mathfrak{A}_{\mathcal{T}}$) is bounded exponentially in $|\mathcal{K}|$. Moreover, $\mathfrak{A}_{\mathcal{T}}$ can be constructed in double exponential time in $|\mathcal{K}|$ since the size of the alphabet and the number of states are exponentially bounded in $|\mathcal{K}|$.

Having Lemma 3 above at hand, it is routine to show the correctness of the constructed automaton. Indeed, $\mathfrak{A}_{\mathcal{T}}$ basically implements the 'strategy' provided by this lemma. □

H Query Automaton $\mathfrak{A}_{\varphi}$

We first prove the characterisation lemma.

Lemma 5. *A function $\pi : \mathbf{x} \cup I_{\varphi} \to \Delta^{\mathcal{I}_{(T,\tau)}}$ with $\pi(a) = [\varepsilon]_a$, for every $a \in I_{\varphi}$, is a match for φ in $\mathcal{I}_{(T,\tau)}$ iff there is a q_i such that for every $\mathfrak{B}(t,t')$ in $\hat{q}_i$, there is a* witness sequence

$$(d_0, s_0), w_1, (d_1, s_1), w_2, \ldots, w_n, (d_n, s_n),$$

where $(d_i, s_i) \in \Delta \times S_{\mathfrak{B}}$ and $w_i \in T$ and such that:

(a) $s_0 = s_{0\mathfrak{B}}$, $s_n \in F_{\mathfrak{B}}$,

(b) $\pi(t) = [w_1]_{d_0}$, $\pi(t') = [w_n]_{d_n}$, *and*

(c) *for every $i \in [1, n]$, we have $d_{i-1}, d_i \in \Delta^{\mathcal{I}_{w_i}}$, $w_i \in [w_{i-1}]_{d_{i-1}}$ if $i \neq 1$, and $\mathcal{I}_{w_i} \models \mathfrak{B}_{s_{i-1},s_i}(d_{i-1}, d_i)$.*

Proof. ($\Rightarrow$) Let π be a match for q in $\mathcal{I}_{(T,\tau)}$, and let $\mathfrak{B}(t,t') \in \hat{q}$. We construct a sequence as required.

By definition of a match, we know that $\mathcal{I}_{(T,\tau)}, \pi \models \mathfrak{B}(t,t')$, that is, there is a word $\nu_1 \cdots \nu_n \in L(\mathcal{E})$ and a sequence $[w_0]_{d_0}, \ldots, [w_n]_{d_n} \in \Delta^{\mathcal{I}_{(T,\tau)}}$ such that $[w_0]_{d_0} = \pi(t), [w_n]_{d_n} = \pi(t')$, and for all $i \in [1, n]$ we have that

(i) if $\nu_i = A?$, then $[w_{i-1}]_{d_{i-1}} = [w_i]_{d_i} \in A^{\mathcal{I}_{(T,\tau)}}$,

(ii) if $\nu_i = r$, then $([w_{i-1}]_{d_{i-1}}, [w_i]_{d_i}) \in r^{\mathcal{I}_{(T,\tau)}}$, and

(iii) if $\nu_i = r^-$, then $([w_i]_{d_i}, [w_{i-1}]_{d_{i-1}}) \in r^{\mathcal{I}_{(T,\tau)}}$.

Observe now that the replacement of r and r^- by $r \cdot r^*$ and $r^- \cdot (r^-)^*$, respectively, for all transitive roles together with the definition of encoding implies that we can assume without loss of generality that for all $i \in [1, n]$, we have:

(i)' if $\nu_i = A?$, then $d_{i-1} = d_i$, and $d_i \in A^{\mathcal{I}_{w_i}}$,

(ii)' if $\nu_i = r$, then $(d_{i-1}, d_i) \in r^{\mathcal{I}_{w_i}}$, and

(iii)' if $\nu_i = r^-$, then $(d_i, d_{i-1}) \in r^{\mathcal{I}_{w_i}}$.

Moreover, there is a sequence of states $s_0, \ldots, s_n \in Q_{\mathfrak{B}}$ such that $s_0 = s_{0\mathfrak{B}}$, $s_n \in F_{\mathfrak{B}}$ and $(s_i, \nu_i, s_{i+1}) \in \Delta_{\mathfrak{B}}$, for all $i \in [0, n-1]$. Thus, the sequence $(d_0, s_0), w_1, \ldots, (d_{n-1}, s_{n-1}), w_n, (d_n, s_n)$ satisfies Items *(a)*–*(c)* of the Lemma.

($\Leftarrow$) Assume that the sequences exist for every $\mathfrak{B}(t,t') \in \hat{q}$. We show that π is a match. Let $(d_0, s_0), w_1, \ldots, w_n, (d_n, s_n)$ be the sequence for some $\mathfrak{B}(t,t') \in \hat{q}$. By Item *(c)*, we obtain

$$\mathfrak{I}(w_1) \models \mathfrak{B}_{s_0,s_1}(d_0, d_1), \ldots, \mathfrak{I}(w_n) \models \mathfrak{B}_{s_{n-1},s_n}(d_{n-1}, d_n).$$

Since $(T, \mathfrak{I}, \mathfrak{r})$ is a tree decomposition of $\mathcal{I}$, we also have

$$\mathcal{I} \models \mathfrak{B}_{s_0,s_1}(d_0, d_1), \ldots, \mathcal{I} \models \mathfrak{B}_{s_{n-1},s_n}(d_{n-1}, d_n).$$

This implies $\mathcal{I} \models \mathfrak{B}_{s_0,s_n}(d_0, d_n)$ and, by Item *(a)*, $\mathcal{I} \models \mathfrak{B}(d_0, d_n)$. Finally, using Item *(b)*, we obtain $\mathcal{I}, \pi \models \mathfrak{B}(t,t')$. □

The following lemma provides a crucial observation underlying the design (and correctness) of the automaton.

Lemma 11. *Let $(d_0, s_0), w_1, \ldots, w_n, (d_n, s_n)$ be a witness sequence satisfying (a)–(c) from Lemma 5, and $i < j$. If $[w_i]_{d_i} \cap [w_j]_{d_j} \neq \emptyset$, then either $j = i + 1$ or there is an $i < m < j$ such that $[w_m]_{d_m} \cap [w_i]_{d_i} \cap [w_j]_{d_j} \neq \emptyset$.*

Proof. If $j = i + 1$, we are done. So assume that $j > i + 1$, and define sets $W_i = [w_i]_{d_i}$ and $W_j = [w_j]_{d_j}$, and

$$V = \bigcup_{i<k<j} [w_k]_{d_k}.$$

Note that each of W_i and W_j are connected subsets of T. By Condition *(c)*, also V is connected. Moreover, by assumption $W_i \cap W_j \neq \emptyset$, and, again by Condition *(c)*, both $W_i \cap V \neq \emptyset$ and $W_j \cap V \neq \emptyset$. Since they are subsets of a tree, their joint intersection $W_i \cap W_j \cap V$ cannot be empty. Hence, there is an m as required. □

Lemma 6. *There is 2ATA $\mathfrak{A}_\varphi$ such that for every $(T,\tau) \in L(\mathfrak{A}_{\mathsf{can}})$, we have $(T,\tau) \in L(\mathfrak{A}_\varphi)$ iff $\mathcal{I}_{(T,\tau)} \models \varphi$. It can be constructed in exponential time in $|\varphi| + |\mathcal{K}|$ and has exponentially in $|\varphi| + |\mathcal{K}|$ many states.*

For the construction of the automaton $\mathfrak{A}_\varphi$, recall that k is the outdegree underlying the input trees (T,τ) and that Δ is the finite domain (of size $2K$). We construct the 2ATA $\mathfrak{A}_\varphi = (Q, \Sigma, q_0, \delta, F)$ as follows. Recall that φ can be equivalently rewritten into a disjunction $q_1 \vee \ldots \vee q_m$ of CRPQs. Slightly abusing notation, we sometimes treat the q_i as sets of atoms.

Set of states. States in Q take four forms. The basic states are q_0, and for each $1 \leq i \leq m$, the CRPQ $\hat{q}_i$. States of the third form are all tuples $\langle p, V_l, V_r\rangle$ such that there is a i such that

- $p \subseteq \hat{q}_i$ and $I_p = \emptyset$,
- V_l is a set of expressions $(d,s) \to_{\mathfrak{B}} x$ such that $\mathfrak{B}$ is the automaton of some atom $\mathfrak{B}(t,t')$ in $\hat{q}_i$, $s \in Q_{\mathfrak{B}}$, $d \in \Delta$, $x \in \mathsf{var}(p)$, and for each $\mathfrak{B}(t,t')$ in φ, there is at most one such expression,
- V_r is a set of expressions $x \to_{\mathfrak{B}} (d,s)$ such that $\mathfrak{B}$ is the automaton of some atom $\mathfrak{B}(t,t')$ in $\hat{q}_i$, $s \in Q_{\mathfrak{B}}$, $d \in \Delta$, $x \in \mathsf{var}(p)$, and for each $\mathfrak{B}(t,t')$ in φ, there is at most one such expression.

States of the fourth form are tuples $\langle d, s, \mathfrak{B}, d', s'\rangle$ with $d, d' \in \Delta$, $s, s' \in Q_{\mathfrak{B}}$, and $\mathfrak{B}$ in φ.

Intuitively, a state of form $\langle p, V_l, V_r\rangle$ expresses the following obligations:

- each atom $\mathfrak{B}(t,t')$ in p still has to be 'processed',
- each $(d,s) \to_{\mathfrak{B}} x$ means that we need to find a path from s to a final state in $\mathfrak{B}$ which is also a path from d to the image of variable x,
- each $x \to_{\mathfrak{B}} (d,s)$ means that we need to find a path from $q_{0\mathfrak{B}}$ to s in $\mathfrak{B}$ which is also a path from the image of variable x to d.

A state of the second form $\langle d, s, \mathfrak{B}, d', s'\rangle$ expresses the obligation that there is a path along which we can reach both d' from d in $\mathcal{I}_{(T,\tau)}$ and s' from s in $\mathfrak{B}$.

Transition function. As a general proviso, we set $\delta(q, \bullet) = \mathsf{false}$, for all states $q \in Q$; in what follows, we define the transitions only for symbols of the form $\sigma = (\mathcal{I}, x) \in \Sigma$. As the transition function does not depend on x, we generally write only $\mathcal{I}$.

The automaton starts off in state q_0 by choosing non-deterministically a disjunct q_i:

$$\delta(q_0, \mathcal{I}) = \bigvee_{1 \leq i \leq m} (0, \hat{q}_i).$$

For every state $\hat{q}_i$, we define a transition as follows. Let $\Theta(\hat{q}_i)$ be the set of all triples (Q_0, V_l, V_r) which can be the result of the following procedure:

1. initialize $V_l = V_r = Q_0 := \emptyset$;
2. for each $\mathfrak{B}(a,b) \in \hat{q}_i$, choose some $s_f \in F_{\mathfrak{B}}$ and add $\langle a, s_{\mathfrak{B}0}, \mathfrak{B}, b, s_f\rangle \in Q_0$;
3. for all $\mathfrak{B}(x,a) \in \hat{q}_i$, choose some $s_f \in F_{\mathfrak{B}}$ and add $x \to_{\mathfrak{B}} (a, s_f) \in V_r$;
4. for all $\mathfrak{B}(a,x) \in \hat{q}_i$, add $(a, s_{0\mathfrak{B}}) \to_{\mathfrak{B}} x \in V_l$.

Intuitively, we choose an accepting state in $F_{\mathfrak{B}}$ for every occurrence of an individual name as the second argument in some atom $\mathfrak{B}(t,t')$. Moreover, obtain p from $\hat{q}_i$ by dropping all atoms mentioning an individual name. The transition for $\hat{q}_i$ is then

$$\delta(\hat{q}_i, \mathcal{I}) = \bigvee_{(Q_0, V_l, V_r) \in \Theta(\hat{q}_i)} \big((0, \langle p, V_l, V_r\rangle) \wedge \bigwedge_{q \in Q_0} (0, q)\big).$$

For transitions for states of the form $\langle d, s, \mathfrak{B}, d', s'\rangle$, we take inspiration from Lemma 11. We start with setting $\delta(\langle d, s, \mathfrak{B}, d', s'\rangle, \mathcal{I}) = \mathsf{false}$ whenever $\{d, d'\} \not\subseteq \Delta^{\mathcal{I}}$, and assume from now on that $\{d, d'\} \subseteq \Delta^{\mathcal{I}}$. The base case is the following:

$$\delta(\langle d, s, \mathfrak{B}, d', s'\rangle, \mathcal{I})) = \mathsf{true} \quad \text{if } \mathcal{I} \models \mathfrak{B}_{s,s'}(d, d').$$

For the case when $\mathcal{I} \not\models \mathfrak{B}_{s,s'}(d,d')$, we include the following transitions:

$$\delta(\langle d, s, \mathfrak{B}, d', s'\rangle, \mathcal{I}) = \bigvee_{i \in [k]} (i, \langle d, s, \mathfrak{B}, d', s'\rangle) \vee$$
$$\bigvee_{\substack{d'' \in \Delta^{\mathcal{I}},\\ s'' \in Q_{\mathfrak{B}}}} \big((0, \langle d, s, \mathfrak{B}, d'', s''\rangle) \wedge (0, \langle d'', s'', \mathfrak{B}, d', s'\rangle)\big)$$

Intuitively, the automaton looks for a node to continue (first line) and then intersects the path non-deterministically (second line, c.f., Lemma 11); it is successful if it finds a node where the required path exists inside the associated interpretation (base case).

For states of the form $\langle p, V_l, V_r\rangle$, we start with including the transitions $\delta(\langle p, V_l, V_r\rangle, \bullet) = \mathsf{false}$ whenever $p = V_l = V_r = \emptyset$, and

$$\delta(\langle p, V_l, V_r\rangle, \mathcal{I}) = \mathsf{false}$$

whenever there is a $(d,s) \to_{\mathcal{A}} x \in V_l$ or a $x \to_{\mathcal{E}} (d,s) \in V_r$ with $d \notin \Delta^{\mathcal{I}}$.

So assume now that $\langle p, V_l, V_r\rangle$ and $\mathcal{I}$ are compatible in this sense, and let $S = \mathsf{var}(p) \cup \{x \mid (d,s) \to_{\mathfrak{B}} x \in V_l\} \cup \{x \mid x \to_{\mathfrak{B}} (d,s) \in V_r\}$. We denote with $\mathcal{P}(S,k)$ the set of all partitions of S into $k+1$ pairwise disjoint, possibly empty sets $S_0, \ldots, S_k$. For each $\mathbf{S} = (S_0, \ldots, S_k) \in \mathcal{P}(S,k)$, define $\Theta(p, V_l, V_r, \mathbf{S})$ as the set of all tuples $(Q^0, p^1, V_l^1, V_r^1, \ldots, p^k, V_l^k, V_r^k)$ that can be obtained as the result of the following procedure.

1. for every $x \in S_0$, choose a value $d_x \in \Delta^{\mathcal{I}}$;
2. for every atom $\mathfrak{B}(x,y) \in p$ with $\{x,y\} \subseteq S$ choose $s_f \in F_{\mathcal{E}}$ and add $\langle d_x, s_{0\mathfrak{B}}, \mathfrak{B}, d_y, s_f\rangle$ to Q^0;
3. for every atom $\mathfrak{B}(x,y) \in p$ with $x \in S_0$, $y \in S_i$ for $i > 0$, choose a value $d_{\mathfrak{B}y} \in \Delta^{\mathcal{I}}$ and a state $s_y \in Q_{\mathfrak{B}}$, and add $\langle d_x, s_{0\mathfrak{B}}, \mathfrak{B}, d_{\mathfrak{B}y}, s_y\rangle$ to Q^0 and $(d_{\mathfrak{B}y}, s_y) \to_{\mathfrak{B}} y$ to V_l^i;
4. for every atom $\mathfrak{B}(x,y) \in p$ with $y \in S_0$, $x \in S_i$ for $i > 0$, choose a value $d_{\mathfrak{B}x} \in \Delta^{\mathcal{I}}$ and states $s_x \in Q_{\mathfrak{B}}$, $s_f \in F_{\mathfrak{B}}$, and add $\langle d_{\mathfrak{B}x}, s_x, \mathfrak{B}, d_y, s_f\rangle \in Q^0$ and $x \to_{\mathfrak{B}} (d_{\mathfrak{B}x}, s_x) \in V_r^i$;

5. for every atom $\mathfrak{B}(x,y) \in p$ with $x,y \in S_i$ for $i > 0$, add $\mathfrak{B}(x,y)$ to p^i;
6. for every atom $\mathfrak{B}(x,y) \in p$ with $x \in S_i, y \in S_j$ for $i \neq j$ and $i,j > 0$, choose values $d_{\mathfrak{B}x}, d_{\mathcal{E}y} \in \Delta^{\mathcal{I}}$ and states s_x, s_y, and add $\langle d_{\mathfrak{B}x}, s_x, \mathfrak{B}, d_{\mathfrak{B}y}, s_y\rangle \in Q^0$, $(d_{\mathfrak{B}y}, s_y) \rightarrow_{\mathfrak{B}} y \in V_l^j$, and $x \rightarrow_{\mathfrak{B}} (d_{\mathfrak{B}x}, s_x) \in V_r^i$;
7. for every $(d,s) \rightarrow_{\mathfrak{B}} x \in V_l$:
   - if $x \in S_0$, choose some $s_f \in F_{\mathfrak{A}}$ and add $\langle d, s, \mathfrak{A}, d_x, s_f\rangle \in Q^0$;
   - if $x \in S_i$ for $i > 0$, then choose $d' \in \Delta^{\mathcal{I}}$ and $s' \in Q_{\mathfrak{B}}$, and add $\langle d, s, \mathfrak{B}, d', s'\rangle \in Q^0$ and $(d', s') \rightarrow_{\mathfrak{B}} x \in V_l^i$;
8. for every $x \rightarrow_{\mathfrak{B}} (d,s) \in V_r$:
   - if $x \in S_0$, add $\langle d_x, s_{0\mathfrak{B}}, \mathfrak{B}, d, s\rangle \in Q^0$;
   - if $x \in S_i$ for $i > 0$, then choose $d' \in \Delta^{\mathcal{I}}$ and $s' \in Q_{\mathfrak{B}}$, and add $\langle d', s', \mathfrak{B}, d, s\rangle \in Q^0$ and $x \rightarrow_{\mathfrak{B}} (d', s') \in V_r^i$.

We then include the following transition: $\delta(\langle p, V_l, V_r\rangle, \mathcal{I})$ as the following expression:

$$\delta(\langle p, V_l, V_r\rangle, \mathcal{I}) = \bigvee_{\substack{\mathbf{S} \in \mathcal{P}(S,k),\\ (Q^0, p^1, V_l^1, V_r^1, \ldots, p^k, V_l^k, V_r^k) \in \Theta(p, V_l, V_r, \mathbf{S})}} \delta^*$$

where δ^* abbreviates

$$\bigwedge_{q \in Q^0} (0, q) \wedge \bigwedge_{i=1}^{k} (i, \langle p^i, V_l^i, V_r^i\rangle).$$

Finally, we define the parity acceptance condition as $F = Q$ to enforce that no state appears infinitely often.

It should be clear that the number of states of the automaton is bounded by an exponential in $|\varphi|$ and polynomially in Δ, that is, exponentially in $|\mathcal{K}|$. Moreover, it is easy to verify that δ can also be computed in exponential time. To finish the proof of Lemma 6, it remains to show correctness of the constructed automaton.

Lemma 12. *For every $(T,\tau) \in L(\mathfrak{A}_{\mathsf{can}})$, we have that $(T,\tau) \in L(\mathfrak{A}_\varphi)$ iff $\mathcal{I}_{(T,\tau)} \models \varphi$.*

Proof. ($\Rightarrow$) Assume some accepting run of $\mathfrak{A}_\varphi$ on (T,τ). Let $\hat{q}_i$ be the successor state of q_0 in the accepting run. Moreover, define a mapping π by taking:

- $\pi(a) = [\varepsilon]_a$ for all $a \in \mathsf{ind}(q_i)$;
- $\pi(x) = [w]_d$, if the automaton visits w in some state $\langle p, V_l, V_r\rangle$ and selects S_0 with $x \in S_0$ and $d_x = d$.

Note that w and d are uniquely defined by the construction of $\mathfrak{A}_\varphi$. In particular, the definition of the transitions for states of the form $\langle p, V_l, V_r\rangle$ makes sure that each variable x is instantiated precisely once, and thus in a unique world w_x to a unique value d_x. We show how to read off from the accepting run witnessing sequences for every $\mathfrak{B}(t,t') \in \hat{q}_i$. By Lemma 5, this implies that π is a match for q_i (and thus for φ) in $\mathcal{I}_{(T,\tau)}$.

Throughout the construction we maintain the following invariant:

($*$) if $(d_0, s_0), w_1, \ldots, (d_n, s_n)$ is the currently constructed sequence for $\mathfrak{B}(t,t')$, then it satisfies *(a)* and *(b)*. Moreover, the automaton visits w_i in state $(d_{i-1}, s_{i-1}, \mathfrak{B}, d_i, s_i)$, for all $i \in [1,n]$.

Fix some $\mathfrak{B}(t,t') \in \hat{q}_i$. We first distinguish cases on whether or not t, t' are constant names.

- If both t, t' are constant symbols, then the automaton visits ε in state $\langle t, s_{0\mathfrak{B}}, \mathfrak{B}, t', s_f\rangle$ for some $s_f \in F_{\mathfrak{B}}$. We initialize the sequence for $\mathfrak{B}(t,t')$ with $(t, s_{0\mathfrak{B}}), \varepsilon, (t', s_f)$. Obviously, ($*$) is satisfied.
- If t is a constant name and t' is not, then by construction of $\mathfrak{A}_\varphi$, in particular the treatment of V_l in the definition of Θ, there is a sequence $(d_0, s_0), w_1, (d_1, s_1), \ldots, (d_n, s_n)$ such that $d_0 = t, s_0 = s_{0\mathfrak{B}}$, $d_n = d_x$, $w_n = w_x$, $s_n \in F_{\mathfrak{B}}$, and which additionally satisfies ($*$).
- The case that t' is a constant name and t is not is analogous (using V_r).

Now, take an atom $\mathfrak{B}(x,y) \in \hat{q}_i$ with $x, y \in \mathsf{var}(\hat{q}_i)$. By definition of $\mathfrak{A}_\varphi$, there is a unique world w and a state $\langle p, V_l, V_r\rangle$ with $\mathfrak{B}(x,y)$ such that the automaton visits w in state $\langle p, V_l, V_r\rangle$ and selects $\mathfrak{S}$ with $x \in S_i$ and $y \in S_j$ for $i \neq j$. Thus, one of the cases 3, 4, or 6 applies. We distinguish cases:

- In case of 3, by the treatment of V_l, there is a sequence $(d_0, s_0), w_1, (d_1, s_1), \ldots, (d_n, s_n)$ such that $d_0 = t, s_0 = s_{0\mathfrak{B}}$, $d_n = d_x$, $w_n = w_x$, $s_n \in F_{\mathfrak{B}}$, and which additionally satisfies ($*$).
- The case of 4 is analogous.
- In case of 6, we know that $x \in S_i, y \in S_j$ for some $i, j > 0$ and $i \neq j$. By construction, there are $d_{\mathfrak{B}x}, d_{\mathfrak{B}y} \in \Delta^{\mathcal{I}_w}$ and states s_x, s_y and:
  1. by the treatment of V_r in Θ: a sequence $(d_0, s_0), w_1, \ldots, w_n, (d_n, s_n)$ with $d_0 = d_x$, $s_0 = s_{0\mathfrak{B}}$, $w_1 = w_x$, $d_n = d_{\mathfrak{B}x}$, $s_n = s_x$, and such that, for each i, the automaton visits w_i in state $\langle d_{i-1}, s_{i-1}, \mathfrak{B}, d_i, s_i\rangle$;
  2. by the treatment of V_l in Θ: a sequence $(d'_0, s'_0), w'_1, \ldots, w'_m, (d'_m, s'_m)$ with $d'_0 = d_{\mathfrak{B}y}$, $s'_0 = s_y$, $d'_m = d_y$, and $s'_m \in F_{\mathfrak{B}}$, and and such that, for each i, the automaton visits w'_i in state $\langle d'_{i-1}, s'_{i-1}, \mathfrak{B}, d'_i, s'_i\rangle$.

  We then start with the sequence

  $$(d_0, s_0), \ldots, (d_n, s_n), w, (d'_0, s'_0), \ldots, (d'_m, s'_m).$$

  This sequence satisfies ($*$) because of 1. and 2. above and because the automaton visits w in state $\langle d_n, s_n, \mathfrak{B}, d'_0, s'_0\rangle = \langle d_{\mathfrak{B}x}, s_x, \mathfrak{B}, d_{\mathfrak{B}y}, s_y\rangle$.

Thus, for each $\mathfrak{B}(t,t') \in \hat{q}_i$, we have constructed a sequence satisfying ($*$). Next, we refine these sequences such that they also satisfy *(c)*. Let $(d_{i-1}, s_{i-1}), w_i, (d_i, s_i)$ be an infix of the sequence constructed so far for some $\mathfrak{B}(t,t') \in \hat{q}_i$. By ($*$), we know that the automaton visits w_i in $\langle d_{i-1}, s_{i-1}, \mathfrak{B}, d_i, s_i\rangle$. We distinguish cases:

– If the automaton accepts at this point, the sequence satisfies *(c)* for this i, and we are done.
– If the automaton moves to some neighbor $w_i \cdot j$ with $j \in [k]$, then we replace w_i in the sequence by $w_i \cdot j$. Obviously, invariant $(*)$ is preserved.
– If the automaton applies the intersection transition to $d'' \in \Delta^{\mathcal{I}_{w_i}}$ and $s'' \in Q_{\mathfrak{B}}$, then replace $(d_{i-1}, s_{i-1}), w_i, (d_i, s_i)$ with $(d_{i-1}, s_{i-1}), w_i, (d'', s''), w_i(d_i, s_i)$. Obviously, the invariant $(*)$ remains preserved.

Because of the acceptance condition, the latter two cases apply only finitely often, so the process terminates with a sequence that satisfies *(c)* for all i. By Lemma 5, we know that $\mathcal{I}_{(T,\tau)}, \pi \models \mathfrak{B}(t, t')$.

($\Leftarrow$) As $\mathcal{I}_{(T,\tau)} \models \varphi$, there is a match π for φ in $\mathcal{I}$. Thus, there is some i such that $\mathcal{I}_{(T,\tau)}, \pi \models \mathfrak{B}(t, t')$, for all $\mathfrak{B}(t, t') \in \hat{q}_i$. By Lemma 5, there are witnessing sequences for each $\mathfrak{B}(t, t') \in \hat{q}_i$ satisfying Conditions *(a)*–*(c)*. Guided by these sequences, we construct an accepting run of $\mathcal{A}_\varphi$. Throughout the construction of this run, some invariants are preserved. First, whenever the automaton visits a node $w \in T$ in a state $\langle p, V_l, V_r\rangle$, then

(I1) $\mathsf{var}(p) \cup \{x \mid (d, s) \to_{\mathfrak{B}} x \in V_l\} \cup \{x \mid x \to_{\mathfrak{B}} (d, s) \in V_r\}$ is the set of all variables x such that the image of x under π is below or in w;
(I2) $(d, s) \to_{\mathfrak{B}} x \in V_l$ implies $d \in \Delta^{\mathcal{I}_w}$ and, in the sequence for $\mathfrak{B}$, there is some i such that $d_i = d$, $s_i = s$ and for all $j \geq i$, w_j is below or equal w;
(I3) $x \to_{\mathfrak{B}} (d, s) \in V_r$ implies $d \in \Delta^{\mathcal{I}_w}$ and in the sequence for $\mathfrak{B}$, there is some i such that $d_i = d$, $s_i = s$ and for all $j \leq i$, w_j is below or equal w.

Moreover, if the automaton visits a node $w \in T$ in state $\langle d, s, \mathfrak{B}, d', s'\rangle$ then

(I4) $d, d' \in \Delta^{\mathcal{I}_w}$ and there are $i < j$ such that, in the witnessing sequence for $\mathfrak{B}$, we have $d_i = d$, $s_i = s$, $d_j = d'$, $s_j = j$, and $w \in [w_i]_d \cap [w_j]_{d'}$.

Throughout the definition of the run, we use $s_{\mathfrak{B}f}$ to refer to the state s_n in the witnessing sequence for each $\mathfrak{B}(t, t') \in \hat{q}_i$.

The automaton starts in state q_0 and chooses to proceed in $\hat{q}_i$. Define Q_0, V_l, V_r by taking

$$Q_0 = \{\langle a, s_{0\mathfrak{B}}, \mathfrak{B}, b, s_{\mathfrak{B}f}\rangle \mid \mathfrak{B}(a, b) \in \hat{q}_i\}$$
$$V_l = \{(a, s_{0\mathfrak{B}}) \to_{\mathfrak{B}} x \mid \mathfrak{B}(a, x) \in \hat{q}_i\}$$
$$V_r = \{x \to_{\mathfrak{B}} (a, s_{\mathfrak{B}f}) \mid \mathfrak{B}(x, a) \in \hat{q}_i\}$$

and extend the run according to this (possible) choice of Q_0, V_l, V_r. The invariants are obviously true after these first transitions. Assume now that $\mathfrak{A}_\varphi$ visits w in state $\langle p, V_l, V_r\rangle$ and let $S = \mathsf{var}(p) \cup \{x \mid (d, s) \to_{\mathfrak{B}} x \in V_l\} \cup \{x \mid x \to_{\mathfrak{B}} (d, s) \in V_r\}$. First, define a partition $S_0, \ldots, S_k$ as follows: S_0 contains all $x \in S$ such that $\pi(x) = [w]_d$ for some $d \in \Delta^{\mathcal{I}_w}$; denote this witness d with d_x. S_i contains all $x \in S$ such that $\pi(x) = [v]_d \in \Delta^{\mathcal{I}_v}$ for some v in the subtree rooted at $w \cdot i$. Then, define a tuple $(Q^0, p^1, V_l^1, V_r^1 \ldots, \ldots, V_l^k, V_r^k)$ as follows:

– For $\mathfrak{B}(x, y) \in p$ with $\{x, y\} \subseteq S_0$, add $\langle d_x, s_{0\mathfrak{B}}, \mathfrak{B}, d_y, s_{\mathfrak{B}f}\rangle$ to Q^0.
– Let $\mathfrak{B}(x, y) \in p$ with $x \in S_0, y \in S_i$ for $i > 0$. Read off from the witnessing sequence for $\mathfrak{B}(x, y)$ the maximal ℓ such that $w_j = w$ for all $j \leq \ell$, and add $\langle d_x, s_{\mathfrak{B}0}, \mathfrak{B}, d_{\ell+1}, s_{\ell+1}\rangle \in Q^0$ and $(d_{\ell+1}, s_{\ell+1}) \to_{\mathfrak{B}} y \in V_l^i$.
– Let $\mathfrak{B}(x, y) \in p$ with $y \in S_0$, $x \in S_i$ for $i > 0$. Read off from the witnessing sequence for $\mathfrak{B}(x, y)$ the minimal ℓ such that $w_j = w$ for all $j \geq \ell$, and add $\langle d_\ell, s_\ell, \mathfrak{B}, d_y, s_n\rangle \in Q^0$ and $x \to_{\mathfrak{B}} (d_\ell, s_\ell) \in V_r^i$.
– Let $\mathfrak{B}(x, y) \in p$ with $x, y \in S_i$ for $i > 0$. Then add $\mathfrak{B}(x, y) \in p^i$.
– Let $\mathfrak{B}(x, y) \in p$ with $x \in S_i, y \in S_j$ for $i, j > 0$ and $i \neq j$. By definition of S_i and S_j, there have to be indices $l < u \in [0, n]$ such that w_m is below w, for all $m < l$ and all $m \geq l$. Add $\langle d_l, s_l, \mathfrak{B}, d_u, s_u\rangle$ to Q^0, $(d_{d_u, s_u}) \to_{\mathfrak{B}} y$ to V_l^j, and $x \to_{\mathfrak{B}} (d_l, s_l)$ to V_r^i.
– Let $(d, s) \to_{\mathfrak{B}} x \in V_l$. We distinguish two cases:
  – if $x \in S_0$, then add $\langle d, s, \mathfrak{B}, d_x, d_{\mathfrak{B}f}\rangle$ to Q^0;
  – if $x \in S_i$ for $i > 0$, then by invariant (I2), there is some ℓ such that $d_\ell = d$, $s_\ell = s$, and for all $j > \ell$, w_j is below or equal w. By Condition *(c)*, there have to be $j > \ell$ and a successor v of w such that $d_j \in \Delta^{\mathcal{I}_w} \cap \Delta^{\mathcal{I}_v}$, and for all $j' > j$, we have that w_j is below or equal v. Add $\langle d, s, \mathfrak{B}, d_j, s_j\rangle$ to Q^0 and $(d_j, s_j) \to_{\mathfrak{B}} x$ to V_l^i.
– Let $x \to_{\mathfrak{B}} (d, s) \in V_r$. We distinguish two cases:
  – if $x \in S_0$, then add $\langle d_x, d_{\mathfrak{B}0}, \mathfrak{B}, d, s\rangle$ to Q^0;
  – if $x \in S_i$ for $i > 0$, then by invariant (I3), there is some ℓ such that $d_\ell = d$, $s_\ell = s$, and for all $j < \ell$, w_j is below or equal w. By Condition *(c)*, there have to be $j < \ell$ and a successor v of w such that $d_j \in \Delta^{\mathcal{I}_w} \cap \Delta^{\mathcal{I}_v}$, and for all $j' < j$, we have that w_j is below or equal v. Add $\langle d_j, s_j, \mathfrak{B}, d, s\rangle$ to Q^0 and $(d_j, s_j) \to_{\mathfrak{B}} x$ to V_r^i.

We then extend the run by the constructed tuple in the non-deterministic choice in the definition of $\delta(\langle p, V_l, V_r\rangle, (\mathcal{I}, x))$. It is routine to verify that the invariants remain true.

It thus remains to show how to complete the run when the automaton visits a node w in state $\langle d, s, \mathfrak{B}, d', s'\rangle$. By (I4), we know that there are $i < j$ such that, in the witnessing sequence for $\mathfrak{B}$, we have $d_i = d$, $s_i = s$, $d_j = d'$, $s_j = j$, and $w \in [w_i]_{d_i} \cap [w_j]_{d_j}$.

By Lemma 11, either $j = i + 1$ or there is an $i < m < j$ such that there is some $\hat{w} \in [w_m]_{d_m} \cap [w_i]_{d_i} \cap [w_j]_{d_j}$. In the first case, we extend the run such that the automaton visits w_j in $\langle d, s, \mathfrak{B}, d', s'\rangle$ and accepts, because of (c). Otherwise, we extend the run by navigating the automaton in state $\langle d, s, \mathfrak{B}, d', s'\rangle$ to node $\hat{w}$, and apply the intersection transition to $d'' = d_m$ and $s'' = s_m$. It should be clear that (I4) remains preserved.

Since the witness sequences are finite, this process terminates after a finite number of steps and the constructed run thus satisfies the parity condition. □