Journal of Artificial Intelligence Research 78 (2023) 385–458 Submitted 12/2022; published 10/2023

# How to Tell Easy from Hard: Complexity of Conjunctive Query Entailment in Extensions of $\mathcal{ALC}$

**Bartosz Bednarczyk** BARTOSZ.BEDNARCZYK@CS.UNI.WROC.PL
*Computational Logic Group,*
*TU Dresden, Germany*
*Faculty of Mathematics and Computer Science,*
*University of Wrocław, Poland*

**Sebastian Rudolph** SEBASTIAN.RUDOLPH@TU-DRESDEN.DE
*Computational Logic Group,*
*TU Dresden, Germany*
*Center for Scalable Data Analytics and Artificial Intelligence,*
*Dresden/Leipzig, Germany*

## Abstract

It is commonly known that the conjunctive query entailment problem for certain extensions of (the well-known ontology language) $\mathcal{ALC}$ is computationally harder than their knowledge base satisfiability problem while for others the complexities coincide, both under the standard and the finite-model semantics. We expose a uniform principle behind this divide by identifying a wide class of (finitely) locally-forward description logics, for which we prove that (finite) query entailment problem can be solved by a reduction to exponentially many calls of the (finite) knowledge base satisfiability problem. Consequently, our algorithm yields tight ExpTime upper bounds for locally-forward logics with ExpTime-complete knowledge base satisfiability problem, including logics between $\mathcal{ALC}$ and $\mu\mathcal{ALCH}b_{\mathsf{reg}}\mathcal{Q}$ (and more), as well as $\mathcal{ALCSCC}$ with global cardinality constraints, for which the complexity of querying remained open. Moreover, to make our technique applicable in future research, we provide easy-to-check sufficient conditions for a logic to be locally-forward based on several novel versions of the model-theoretic notion of unravellings.

Together with existing results, this provides a nearly complete classification of the "benign" vs. "malign" primitive modelling features extending $\mathcal{ALC}$, missing out only the $\mathsf{Self}$ operator. We then show a rather surprising result, namely that the conjunctive entailment problem for $\mathcal{ALC}^{\mathsf{Self}}$ is exponentially harder than for $\mathcal{ALC}$. This places the seemingly innocuous $\mathsf{Self}$ operator among the "malign" modelling features, like inverses, transitivity or nominals.

## 1. Introduction

Formal ontologies are of significant importance in artificial intelligence, playing a central role in the Semantic Web, ontology-based information integration, or peer-to-peer data management. In such scenarios, an especially prominent role is played by *description logics* (DLs) (Baader, Horrocks, Lutz, & Sattler, 2017) – a robust family of logical formalisms used to describe ontologies and serving as the logical underpinning of contemporary standardised ontology languages. To put knowledge bases to full use as core part of intelligent information systems, much attention is being devoted to the area of ontology-based data-access, with

*conjunctive queries* (CQs) being employed as a fundamental querying formalism (Ortiz & Simkus, 2012).

In recent years, it has become apparent that various modelling features of DLs affect the complexity of conjunctive query entailment in a rather strong sense. Let us focus on the most popular DL, namely $\mathcal{ALC}$. It was first shown by Lutz (2008a) that CQ entailment is exponentially harder than the consistency problem for $\mathcal{ALC}$ extended with inverse roles ($\mathcal{I}$). Shortly after, a combination of transitivity and role hierarchies ($\mathcal{SH}$) was shown as another culprit of higher worst-case complexity of reasoning (Eiter, Lutz, Ortiz, & Simkus, 2009). Finally, also nominals ($\mathcal{O}$) turned out to be problematic (Ngo, Ortiz, & Simkus, 2016). Nevertheless, there are also more benign DL constructs regarding the complexity of CQ entailment. Examples are counting ($\mathcal{Q}$) (Lutz, 2008a) (the complexity stays the same even for expressive arithmetical constraints (Baader, Bednarczyk, & Rudolph, 2020)), role-hierarchies alone ($\mathcal{H}$) (Eiter, Ortiz, & Simkus, 2012) or a tamed use of higher-arity relations (Bednarczyk, 2021a).

Despite the considerable effort in establishing the complexity of the query entailment problem over $\mathcal{ALC}$ extended with various primitive features, for many of the extensions the precise computational complexity is still unknown. To be more precise, the complexity of CQ entailment problem for $\mathcal{ALC}$ extended with any of safe-boolean role combinations ($b$), transitive closure of roles ($\cdot^+$), regular role expressions ($\cdot_{\mathsf{reg}}$), fixed points ($\mu$), or the self operator ($\mathsf{Self}$) is *not known*. Doubly-exponential upper bounds for querying any of the mentioned logics follow from existing results, *e.g.* from the results on the $\mathcal{Z}$ family of DLs (Calvanese, Eiter, & Ortiz, 2009) or from the guarded-negation fixpoint logic (Bárány, ten Cate, & Segoufin, 2015). However, it is not at all clear whether such complexity bounds are tight. And even more intriguingly, we do not know whether the previously established upper bounds for CQ entailment can be adapted to a slightly more general class of queries, namely to the case of entailment of *unions of* conjunctive queries (UCQs). While it is known that for the case of $\mathcal{ALCH}$, the complexities of querying with CQs and UCQs coincide (Ortiz, 2010, Thm. 6.5.1), the case of $\mathcal{ALCQ}$ is not yet resolved. There is no reason to believe that for some DL the complexity of UCQ entailment differs from the complexity of UCQ entailment, but jumping to the class of even more expressive queries, like positive existential queries (PEQs) or conjunctive regular path queries (CRPQs) results in an increase of complexity (Ortiz & Simkus, 2014, Thm. 1).

The aim of this paper is to provide *robust* answers to the questions above, and more generally, to clarify why the query entailment problem for certain extensions of $\mathcal{ALC}$ is computationally harder than their knowledge base satisfiability problem while for others the complexities coincide. By "robustness", we mean that the developed technique should cover multiple extensions of $\mathcal{ALC}$ in one go, without the need of reproducing nearly-identical proofs for freshly defined logics and should be based on well-established notions from the literature.

### 1.1 Overview of Our Results and Organisation of the Paper

We start by introducing the classes $\mathscr{C}_{\mathsf{lf}}$ and $\mathscr{C}_{\mathsf{flf}}$ of *locally-forward* and *finitely locally-forward* description logics. These are classes of ontology languages extending[1] $\mathcal{ALC}^{\cap}$ in which any description logic $\mathcal{DL} \in \mathscr{C}_{\mathsf{flf}}$ (resp. $\mathcal{DL} \in \mathscr{C}_{\mathsf{lf}}$) enjoys a property that each (finitely) satisfiable

1. The use of a role conjunction operator $^{\cap}$, allowing for specifying that two elements are connected via a conjunction of roles, is essential for our querying algorithm to work. See Preliminaries for the definition.

$\mathcal{DL}$-knowledge-base $\mathcal{K}$ has a (finite) model that locally resembles a forward tree. Afterwards, we revisit a classical algorithm for conjunctive querying $\mathcal{ALCHQ}$ designed by Lutz (2008a) based on the so-called *spoiler technique*, and improve the technique in several ways: (i) our algorithm can be applied to unions of CQs rather than plain CQs, (ii) our algorithm works for any logic $\mathcal{DL} \in \mathscr{C}_{\mathsf{lf}}$, (iii) our algorithm can be applied to the finite query entailment problem for logics from $\mathscr{C}_{\mathsf{flf}}$. Despite the employment of Lutz's technique, most of our proofs are done from scratch in order to adjust them to the new, more abstract and more general, setting. In particular, in the case of finite model reasoning, or in reasoning about logics with global cardinality constraints, the intended models are no longer trees (and hence the proof sketches by Lutz that work for $\mathcal{SHQ}$ cannot be taken for granted in our setting). Moreover, we stress that we stick to the usual definition of conjunctive queries, where only role names and concept names are allowed (in particular we do not allow for complex role expressions). Allowing for the presence of complex roles in the query would change the picture drastically, as happens already for the description logic $\mathcal{S}$: if transitive roles are forbidden in queries, the CQ entailment problem over $\mathcal{S}$-knowledge-bases is ExpTime-complete (Lutz, 2008b, Thm. 1), and coNExpTime-hard otherwise (Eiter et al., 2009, Thm. 2). As a transitive role can be mimicked with a transitive closure of a fresh role, the results on the query entailment problem for $\mathcal{S}$ without transitivity in queries can be reproved in our setting (*i.e.* they follow from our results on the query entailment problem for $\mathcal{ALC}_{\mathsf{reg}}$).

**Theorem 1.1.** *Let $\mathcal{DL}$ be a DL from $\mathscr{C}_{\mathsf{lf}}$ (resp. from $\mathscr{C}_{\mathsf{flf}}$). The problem of (finite) entailment of unions of conjunctive queries over $\mathcal{DL}$-knowledge-bases can be solved by an exponential-time procedure involving exponentially many (finite) satisfiability calls as a subroutine.*

Theorem 1.1 implies a tight ExpTime upper bound for querying for all DLs from $\mathscr{C}_{\mathsf{flf}} \cup \mathscr{C}_{\mathsf{lf}}$ enjoying ExpTime-complete knowledge base satisfiability problems. This yields new tight single exponential-time complexity bounds for the entailment of unions of conjunctive queries over many description logics, including expressive sublogics of $\mathcal{ALCHb}_{\mathsf{reg}}\mathcal{Q}$ and (for several of which only a 2ExpTime upper bound was known, see the discussion on the previous page), or $\mathcal{ALC}$ with very expressive counting (no previous results on querying). We stress that we cover all of these logics with a single proof, in an abstract and fully uniform way.

To apply Theorem 1.1 to freshly discovered logics, one may need to check if a given logic belongs to the classes $\mathscr{C}_{\mathsf{lf}}$ or $\mathscr{C}_{\mathsf{flf}}$. This task is however relatively difficult, due to the quite technical definition of (finitely) locally-forward DLs. To simplify such a process, we propose several sufficient conditions based on novel model-theoretic notions of unravellings. Hence, with a bit of luck, this relatively tedious task boils down to routine proofs involving structural induction. What is more, each of our construction is supplemented with a handy list of properties preserved by our unravellings, that can simplify the reasoning even further.

**Theorem 1.2.** *Let $\mathcal{DL}$ extend $\mathcal{ALC}^{\cap}$. If $\mathcal{DL}$ is* preserved under forward-unravellings *(resp.* scattered forward-unravellings with or without rebalancing*) then $\mathcal{DL} \in \mathscr{C}_{\mathsf{lf}}$ (resp. $\mathcal{DL} \in \mathscr{C}_{\mathsf{flf}}$).*

Theorems 1.1–1.2 provide a nearly complete classification of the "querying-satisfiability trade-off" for $\mathcal{ALC}$ extended with popular primitive[2] features. This is summarised by the

2. We do not consider role axioms like role disjointness or reflexivity from $\mathcal{SROIQ}$ (Horrocks, Kutz, & Sattler, 2006) as *primitives*, as they can be expressed with other features.

table below; references for logics having harder query entailment than satisfiability were given in the introduction. Definitions of standard DL features are given in Preliminaries. The only non-standard feature are $\mathcal{SCC}$ (Baader, 2017), which allows the logic for specifying expressive cardinality constraints by means of Presburger arithmetics, ($\mu$) introduces recursion by means of fixed-points. Despite their quite technical definitions and the fact that we need them only for corollaries, we use fully define them in appendix.

| | Feature $\theta$ with name | $\mathcal{ALC}\theta^{\cap} \in \mathscr{C}_{\mathsf{lf}}$? | SAT=CQEnt? |
|---|---|---|---|
| good | functionality $\mathcal{F}$ and various counting: $\mathcal{N}/\mathcal{Q}/\mathcal{SCC}$ | ✓ [new!] | |
| good | trans. closure $\cdot^*$, regular expr. $\cdot_{\mathsf{reg}}$, fixed-points $\mu$ | | |
| good | role hierar. $\mathcal{H}$, safe boolean comb. of roles $b$ | | |
| bad | inverses of roles $\mathcal{I}$ | ✗ | |
| bad | nominals $\mathcal{O}$ | | |
| bad | transitivity $\mathcal{S}$, complex role inclusions $\mathcal{R}$ | | |
| bad | self-loops Self | ✗ [new!] | |

We stress that the features $\mathcal{F}$, $\mathcal{N}$, $\mathcal{Q}$, $\mathcal{H}$ and $b$ are subsumed by $\mathcal{SCC}$, while $\cdot^*$ and $\cdot_{\mathsf{reg}}$ are subsumed by $\mu$ (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2003, p. 204). Hence, the $\mu\mathcal{ALCSCC}$ is a super-logic that comprises all the positive features together.
The above classification does not settle the case of the Self operator, a modelling feature supported by the OWL 2 Web Ontology Language and the DL $\mathcal{SROIQ}$ (Horrocks et al., 2006). The Self operator allows us to specify the situation when an element is related to it*self* by a binary relationship. Among other things, this allows us to formalise the concept of a "narcissist" with Narcissist $\sqsubseteq \exists \mathit{loves}.\mathsf{Self}$. Due to the simplicity of the Self operator (it only refers to one element), it is easy to accommodate for automata techniques (Calvanese et al., 2009) or consequence-based methods (Ortiz, Rudolph, & Simkus, 2010) and thus, so far, there has been no real indication that the added expressivity provided by Self may change anything, complexity-wise. Arguably, this impression is further corroborated by the observation that Self features in two profiles of OWL 2 (the EL and the RL profile), again without harming tractability (Krötzsch, Rudolph, & Hitzler, 2008).

In Section 5 however, we show a rather surprising result, namely that CQ entailment for $\mathcal{ALC}^{\mathsf{Self}}$ is exponentially harder than for $\mathcal{ALC}$. Hence, it places the seemingly innocuous Self operator among the "malign" modelling features, like ($\mathcal{I}$), ($\mathcal{S}$), or ($\mathcal{O}$).

**Theorem 1.3.** *Conjunctive query entailment over $\mathcal{ALC}^{\mathsf{Self}}$ TBoxes is 2ExpTime-hard.*

Our proof goes via encoding of computation trees of alternating Turing machines working in exponential space and follows the general hardness-proof-scheme by Lutz (2008a). However, to adjust the schema to $\mathcal{ALC}^{\mathsf{Self}}$, novel ideas are required: the ability to speak about self-loops is exploited to produce a single query that traverses trees in a root-to-leaf manner and to simulate disjunction inside conjunctive queries, useful to express that certain paths are repeated inside the tree. Theorem 1.3 can be additionally employed to establish 2ExpTime-hardness of query entailment for the $\mathcal{Z}$ family (a.k.a. $\mathcal{ALCHI}b^{\mathsf{Self}}_{\mathsf{reg}}$) of DLs (Calvanese et al., 2009), which until now has remained open[3] as well as the 2ExpTime-hardness of query-

3. We stress that 2ExpTime hardness of CQ entailment over $\mathcal{Z}$ ontologies *does not* follow from 2ExpTime-hardness of the same problem for $\mathcal{SH}$ since we cannot define in $\mathcal{Z}$ that a given role is transitive nor that it is a transitive closure of another role (to simulate transitivity).

ing ontologies formulated in the fluted guarded fragment (Bednarczyk, 2021a) with equality.[4]

To improve the readability of the paper and keep its length in control, most of our *proofs are written in two different versions*. The main body of the paper provides proof sketches, whose intention is to communicate ideas required to reproduce proofs in a couple of minutes with pen and paper. The online appendix provides their *very* detailed counterparts.

### 1.2 Relationship to Prior Conference Publications

This work is a revised version of our existing papers extended with some new results. Section 3 extends the unpublished results from an arXiV note by the first author (Bednarczyk, 2021b). Some results from Section 4 were first presented as part of joint work with F. Baader (2020) published at ECAI'2020; except a few ideas[5], this material has been completely reworked here. Finally, Section 5 is a full version of our AAAI (2022) paper.

## 2. Preliminaries

We recap the basics on description logics (DLs) (Baader et al., 2017) and query entailment over description logic ontologies (Ortiz & Simkus, 2012). As usual, the letter $\mathbb{N}$ (resp. $\mathbb{N}_+$) denotes the set of (positive) natural numbers. With $\mathbb{Z}_n$ we denote the set $\{0, 1, \ldots, n-1\}$.

### 2.1 Basics on DLs

We fix countably infinite pairwise disjoint sets of *individual names* $\mathbf{N_I}$, *concept names* $\mathbf{N_C}$, and *role names* $\mathbf{N_R}$ and introduce a description logic $\mathcal{ALC}^{\cap}$. Starting from $\mathbf{N_C}$ and $\mathbf{N_R}$, the set $\mathbf{C}$ of $\mathcal{ALC}^{\cap}$ *concepts* is built using the following concept constructors: *negation* ($\neg$), *conjunction* ($\sqcap$), *existential restriction* ($\exists(... \cap \ldots \cap ...)$) and the *bottom concept* ($\bot$), with the grammar:

$$\mathrm{C}, \mathrm{D} \ ::= \ \bot \ \mid \ \mathrm{A} \ \mid \ \neg \mathrm{C} \ \mid \ \mathrm{C} \sqcap \mathrm{D} \ \mid \ \exists(r_1 \cap \ldots \cap r_n).\mathrm{C},$$

where $\mathrm{C}, \mathrm{D} \in \mathbf{C}$, $\mathrm{A} \in \mathbf{N_C}$ and $r_1, r_2, \ldots, r_n \in \mathbf{N_R}$. As convenient abbreviations, we often employ disjunction $\mathrm{C} \sqcup \mathrm{D} \coloneqq \neg(\neg \mathrm{C} \sqcap \neg \mathrm{D})$, universal restriction $\forall(r_1 \cap \ldots \cap r_n).\mathrm{C} \coloneqq \neg\exists(r_1 \cap \ldots \cap r_n).\neg \mathrm{C}$, top concept $\top \coloneqq \neg\bot$, and – not quite as widely used – "inline-implication" $\mathrm{C} \to \mathrm{D} \coloneqq \neg \mathrm{C} \sqcup \mathrm{D}$.

*Assertions* are expressions of the form $\mathrm{C}(\mathsf{a})$, $r(\mathsf{a}, \mathsf{b})$, or $\neg r(\mathsf{a}, \mathsf{b})$ for $\mathsf{a}, \mathsf{b} \in \mathbf{N_I}$, $\mathrm{C} \in \mathbf{C}$ and $r \in \mathbf{N_R}$. A *general concept inclusion* (GCI) has the form $\mathrm{C} \sqsubseteq \mathrm{D}$ for concepts $\mathrm{C}, \mathrm{D} \in \mathbf{C}$. We use $\mathrm{C} \equiv \mathrm{D}$ as a shorthand for the joint occurrence of the two GCIs $\mathrm{C} \sqsubseteq \mathrm{D}$ and $\mathrm{D} \sqsubseteq \mathrm{C}$. A *knowledge base* (KB) $\mathcal{K} \coloneqq (\mathcal{A}, \mathcal{T})$ is composed of a finite set $\mathcal{A}$ (*ABox*) of assertions and a finite non-empty set $\mathcal{T}$ (*TBox*) of GCIs. We call the elements of $\mathcal{A} \cup \mathcal{T}$ *axioms*. The set of all individual names appearing in $\mathcal{K}$ is denoted with $\mathsf{ind}(\mathcal{K})$.

The semantics of $\mathcal{ALC}^{\cap}$ is defined via *interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ composed of a non-empty set $\Delta^{\mathcal{I}}$ called the *domain of* $\mathcal{I}$ and an *interpretation function* $\cdot^{\mathcal{I}}$ mapping individual

4. Self can be expressed with $\forall x_1 \mathrm{self}_{\mathrm{r}}(x_1) \to \exists x_2\,(r(x_1, x_2) \wedge x_1 = x_2) \wedge \forall x_1 \forall x_2 r(x_1, x_2) \to (x_1 = x_2 \to \mathrm{self}_{\mathrm{r}}(x_2))$.
5. Speaking more precisely: the notion of forward unravelling and duplication are nearly the same here as in the ECAI paper (except the fact that here they are presented in the broader context with more properties and full proofs) nearly the same while the notion of loosening is replaced by completely new construction under the name of scattered forward unravellings.

names to elements of $\Delta^{\mathcal{I}}$, concept names to subsets of $\Delta^{\mathcal{I}}$, and role names to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. This mapping is inductively extended to concepts via

$$\begin{aligned} \bot^{\mathcal{I}} &:= \emptyset, \\ (\neg \mathrm{C})^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \setminus \mathrm{C}^{\mathcal{I}}, \\ (\mathrm{C} \sqcap \mathrm{D})^{\mathcal{I}} &:= \mathrm{C}^{\mathcal{I}} \cap \mathrm{D}^{\mathcal{I}}, \\ (\exists (r_1 \cap \ldots \cap r_n).\mathrm{C})^{\mathcal{I}} &:= \{\mathrm{d} \mid \exists \mathrm{e} \in \mathrm{C}^{\mathcal{I}}.\ (\mathrm{d},\mathrm{e}) \in \bigcap_{i=1}^{n} r_i^{\mathcal{I}}\}, \end{aligned}$$

and finally used to define *satisfaction* of assertions and GCIs in an interpretation $\mathcal{I}$ by letting

$$\begin{array}{lll} \mathcal{I} \models \mathrm{C} \sqsubseteq \mathrm{D} & \textit{if and only if} & \mathrm{C}^{\mathcal{I}} \subseteq \mathrm{D}^{\mathcal{I}}, \\ \mathcal{I} \models \mathrm{C}(\mathsf{a}) & \textit{if and only if} & \mathsf{a}^{\mathcal{I}} \in \mathrm{C}^{\mathcal{I}}, \\ \mathcal{I} \models r(\mathsf{a},\mathsf{b}) & \textit{if and only if} & (\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}}) \in r^{\mathcal{I}}, \\ \mathcal{I} \models \neg r(\mathsf{a},\mathsf{b}) & \textit{if and only if} & (\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}}) \notin r^{\mathcal{I}}. \end{array}$$

We define $\mathcal{ALC}$-concepts and knowledge bases analogously, by requiring $n = 1$ in existential restrictions. Given an interpretation $\mathcal{I}$ and a pair of elements $\mathrm{d}, \mathrm{e} \in \Delta^{\mathcal{I}}$, with $\mathsf{Conc}_{\mathcal{I}}(\mathrm{d}) := \{\mathrm{C} \in \mathbf{N_C} \mid \mathrm{d} \in \mathrm{C}^{\mathcal{I}}\}$ we denote the set of all atomic concepts satisfied by d in $\mathcal{I}$, and with $\mathsf{Rol}_{\mathcal{I}}(\mathrm{d},\mathrm{e}) := \{r \in \mathbf{N_R} \mid (\mathrm{d},\mathrm{e}) \in r^{\mathcal{I}}\}$ we denote the set of all atomic roles satisfied by the pair $(\mathrm{d},\mathrm{e})$ in $\mathcal{I}$.

*Structures* are like interpretations, with the only exception that the assignment of individual names may be partial, that is some individual names from $\mathbf{N_I}$ may not be "used". We say that an interpretation $\mathcal{I}$ *satisfies* a KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ (or $\mathcal{I}$ is a *model* of $\mathcal{K}$, written: $\mathcal{I} \models \mathcal{K}$) if it satisfies all axioms of $\mathcal{A} \cup \mathcal{T}$. An interpretation $\mathcal{I}$ is *finite* (resp. countable) iff its domain $\Delta^{\mathcal{I}}$ is finite (resp. countable). A KB is (finitely) *consistent* (or (finitely) *satisfiable*) if it has a (finite) model, and (finitely) *inconsistent* (or (finitely) *unsatisfiable*) otherwise.

Given a set of individual names $\mathsf{N} \subseteq \mathbf{N_I}$ we denote with $\mathsf{N}^{\mathcal{I}}$ the set of $\mathsf{N}$-*named domain elements* of $\mathcal{I}$, *i.e.* the set of all $\mathrm{d} \in \Delta^{\mathcal{I}}$ for which $\mathrm{d} = \mathsf{a}^{\mathcal{I}}$ holds for some name $\mathsf{a} \in \mathsf{N}$. The elements from $\Delta^{\mathcal{I}} \setminus \mathsf{N}^{\mathcal{I}}$ are called $\mathsf{N}$-*anonymous*. We also employ $\mathsf{ind}(\mathcal{I}) := \{\mathsf{a} \in \mathbf{N_I} \mid \mathsf{a}^{\mathcal{I}} \in \Delta^{\mathcal{I}}\}$ to collect all individual names whose interpretation appears in a structure $\mathcal{I}$.

The presented notions are straightforwardly lifted to any description logic $\mathcal{DL}$ semantically extending $\mathcal{ALC}^{\cap}$ and allowing for polynomial expressivity of $\mathcal{ALC}^{\cap}$ concepts, *i.e.* for every $\mathcal{ALC}^{\cap}$-concept C there exists some (at most) polynomially larger, logically equivalent, $\mathcal{DL}$-concept. Throughout the paper, such logics will be called *abstract expressive description logics* or simply *abstract DLs*.[6]

### 2.2 Various Primitive DL Features

We next provide various further DL modelling features mentioned in the introduction. We focus on the most popular ones, whereas the less known features ($\mathcal{SCC}$) and ($\mu$) are defined in appendix.

6. We have decided not to formally define what a *semantic extension* of $\mathcal{ALC}^{\cap}$ is, suggesting that this notion should rather be understood naively. Promising examples of *abstract DLs* are well-known DLs like $\mathcal{ALC}^{\cap}, \mathcal{ALCOIQ}^{\cap}, \mathcal{SHQ}^{\cap}, \mathcal{Z}, \mu\mathcal{ALC}^{\cap}$ etc. Of course, the notion of *abstract DLs* can be made formal, by employing abstract model theory. Consult PhD thesis of Piro (2012, Section 1.2).

1. Functionality ($\mathcal{F}$) allows us to specify, by means of a new axiom type of the form $\mathsf{func}(r)$, that a given role name $r$ must be interpreted as a functional relation, *i.e.* whenever $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$ and $(\mathrm{d}, \mathrm{e}') \in r^{\mathcal{I}}$ holds in an interpretation $\mathcal{I}$ satisfying $\mathsf{func}(r)$, then $\mathrm{e} = \mathrm{e}'$ must hold.
2. Qualified number restrictions ($\mathcal{Q}$). We extend the definition of concepts with constructs of the form $(\geq n\ r).\mathrm{C}$, where $n \in \mathbb{N}$ is a number $r \in \mathbf{N_R}$ is a role name, and C is a concept. Given $\mathcal{I}$, the interpretation of $(\geq n\ r).\mathrm{C}$ is defined as follows:

$$\mathrm{d} \in ((\geq n\ r).\mathrm{C})^{\mathcal{I}} \text{ if and only if } |\{\mathrm{e} \in \Delta^{\mathcal{I}} : (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}} \text{ and } \mathrm{e} \in \mathrm{C}^{\mathcal{I}}\}| \geq n$$

   In unqualified number restrictions ($\mathcal{N}$), the concept C in the expression $(\geq n\ r).\mathrm{C}$ must be $\top$ (and is usually omitted). Note that ($\mathcal{Q}$) can express both ($\mathcal{N}$) and ($\mathcal{F}$).
3. Role hierarchies ($\mathcal{H}$) allow us to specify inclusions between *atomic* roles by means of an extra axiom type of the shape $r \subseteq s$ for *role names* $r, s \in \mathbf{N_R}$. Formally $\mathcal{I} \models (r \subseteq s)$ if and only if $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$.
4. Safe boolean role combinations ($b$) introduce a notion of a *simple role*, that is defined inductively as follows: (i) every role name is simple (ii) intersection, union and set difference of two simple roles is also simple. The semantics of simple roles is defined in an obvious way, employing the usual set-theoretic semantics of $\cup, \cap, \setminus$. Simple roles can then be used in existential and universal restrictions (as well as in the number restrictions whenever they are admitted in the logic considered), replacing the usual notion of roles. Note that $\mathcal{I} \models (\exists(s \setminus r).\top) \sqsubseteq \bot$ is equivalent to $\mathcal{I} \models (r \subseteq s)$, thus ($b$) subsumes ($\mathcal{H}$).
5. The feature ($\mathcal{I}$) introduces, per each role name $r \in \mathbf{N_R}$, a fresh role name $r^{-}$ interpreted via $(r^{-})^{\mathcal{I}} = \{(\mathrm{e}, \mathrm{d}) \mid (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}$.
6. The feature ($\mathcal{O}$) introduces, per each individual name $\mathsf{a} \in \mathbf{N_I}$, a fresh concept $\{\mathsf{a}\}$, interpreted via $\{\mathsf{a}\}^{\mathcal{I}} := \{\mathsf{a}^{\mathcal{I}}\}$.
7. *Transitivity* ($\mathcal{S}$) allows us for specifying, by means of a new axiom type of the form $\mathsf{trans}(r)$, that a given role name $r$ is interpreted as a transitive relation. The feature ($\mathcal{R}$) allows us for specifying *chains of complex role inclusions* (Horrocks et al., 2006, p. 2–3) that generalise transitivity. We assume that if a role name $r$ occurs in a transitivity statement $\mathsf{trans}(r)$ or on the right hand side of a complex role inclusion, then it is disallowed from number restrictions.
8. The feature ($\mathsf{Self}$) introduces concepts of the form $\exists r.\mathsf{Self}$, where $r \in \mathbf{N_R}$. Its semantics is defined as follows: $(\exists r.\mathsf{Self})^{\mathcal{I}} := \{\mathrm{d} \in \mathrm{C}^{\mathcal{I}} \mid (\mathrm{d}, \mathrm{d}) \in r^{\mathcal{I}}\}$.
9. Enriching a description logic with the transitive closure construct $(\cdot^{+})$ gives rise to fresh concepts of the form $\exists r^{+}.\mathrm{C}$ and $\forall r^{+}.\mathrm{C}$, where $r$ is a role name and C is a (possibly complex) concept. The semantics of $\exists r^{+}.\mathrm{C}$ is defined as follows:

$$(\exists r^{+}.\mathrm{C})^{\mathcal{I}} := \Big\{\mathrm{d} \mid \exists \mathrm{e} \in \mathrm{C}^{\mathcal{I}}. \bigvee_{k=1}^{\infty} \Big((\mathrm{d}, \mathrm{e}) \in (r^{\mathcal{I}})^{k}\Big)\Big\},$$

   where $(r^{\mathcal{I}})^{k}$ denotes the $k$-fold composition of $r^{\mathcal{I}}$.

10. The $(\cdot_{\mathsf{reg}})$ feature introduces new concepts of the form $\exists t.\mathrm{C}$ and $\forall t.\mathrm{C}$, where $t$ is a regular role expression defined according to the grammar:

$$t ::= r \in \mathbf{N_R} \mid t \cup t \mid t \circ t \mid t^* \mid id(\mathrm{C}).$$

The use of role names in the above grammar can be replaced by simple roles, whenever the DL feature $(b)$ is included in the language. The semantics of role intersection, union and composition are defined as in the case of $(b)$ and $(\mathcal{R})$, while the reflexive-transitive closure $t^*$ is interpreted in $\mathcal{I}$ as:

$$t^* = \bigcup_{k=0}^{\infty} t^k,$$

where $t^k$ is the $k$-fold composition of $t$. Finally, we define the satisfaction of "concept tests" in $\mathcal{I}$ as $(id(\mathrm{C}))^{\mathcal{I}} := \{(\mathrm{d},\mathrm{d}) \mid \mathrm{d} \in \mathrm{C}^{\mathcal{I}}\}$. Obviously $(\cdot_{\mathsf{reg}})$ subsumes $(\cdot^+)$.

### 2.3 A Tiny Bit of Graph Theory and Dendrology

We revisit the classical notions of substructures, paths and connectivity. The *restriction* of a structure $\mathcal{I}$ to a set $S \subseteq \Delta^{\mathcal{I}}$, is the structure $\mathcal{I}{\restriction_S}$ with $\Delta^{\mathcal{I}{\restriction_S}} := S$ defined by:

$$r^{\mathcal{I}{\restriction_S}} := r^{\mathcal{I}} \cap (S \times S),\ \mathrm{A}^{\mathcal{I}{\restriction_S}} := \mathrm{A}^{\mathcal{I}} \cap S,\ \mathsf{a}^{\mathcal{I}{\restriction_S}} := \mathsf{a}^{\mathcal{I}} \text{ if } \mathsf{a}^{\mathcal{I}} \in S \text{ and undefined otherwise,}$$

for all concept names $\mathrm{A} \in \mathbf{N_C}$, role names $r \in \mathbf{N_R}$ and individual names $\mathsf{a} \in \mathbf{N_I}$. A *substructure* of $\mathcal{I}$ is any of its restrictions $\mathcal{I}{\restriction_S}$ for any sets $S \subseteq \Delta^{\mathcal{I}}$.

The notion of paths is introduced next. An *undirected path* (resp. a *directed path*) of length $k{-}1$ in $\mathcal{I}$ is a word $\rho = \rho_1\rho_2\ldots\rho_k \in (\Delta^{\mathcal{I}})^+$ such that for any index $i < k$ we have that $(\rho_i, \rho_{i+1}) \in r^{\mathcal{I}}$ or $(\rho_{i+1}, \rho_i) \in r^{\mathcal{I}}$ for some role name $r \in \mathbf{N_R}$ (or just $(\rho_i, \rho_{i+1}) \in r^{\mathcal{I}}$ in the directed case). An element $\mathrm{e} \in \Delta^{\mathcal{I}}$ is *reachable* from $\mathrm{d} \in \Delta^{\mathcal{I}}$ via an (un)directed path if there exists an (un)directed path $\rho = \rho_1\rho_2\ldots\rho_k$ in $\mathcal{I}$ with $\rho_1 = \mathrm{d}$ and $\rho_k = \mathrm{e}$. We say that $\mathcal{I}$ is *connected* if any of its domain elements are reachable from any other via an undirected path. A structure $\mathcal{J}$ is a *connected component* of $\mathcal{I}$ if it is a $\subseteq$-maximal (in the sense of inclusion of domains) connected substructure of $\mathcal{I}$. For any number $k \geq 0$ we define the *$k$-neighbourhood* of d in $\mathcal{I}$, denoted with $\mathsf{Nbd}^k_{\mathcal{I}}(\mathrm{d})$, as the restriction of $\mathcal{I}$ to elements reachable from d in $\mathcal{I}$ by *undirected* paths of length $\leq k$.

Given a set $\mathbb{D}$, we say that a structure $\mathcal{I}$ is a *$\mathbb{D}$-forward-forest*, if $\Delta^{\mathcal{I}}$ is a prefix-closed subset of $\mathbb{D}^+$, and for all role names $r \in \mathbf{N_R}$, if $(\mathrm{d},\mathrm{e}) \in r^{\mathcal{I}}$ then either $\mathrm{d},\mathrm{e} \in \mathbb{D}$ or $\mathrm{e} = \mathrm{d}\cdot\mathrm{c}$ for some $\mathrm{c} \in \mathbb{D}$. Here the superscript $^+$ denotes Kleene plus and the $\cdot$ operator (often omitted) denotes word concatenation. The elements of $\Delta^{\mathcal{I}} \cap \mathbb{D}$ are called the *roots* of $\mathcal{I}$. We call $\mathcal{I}$ a *$\mathbb{D}$-forward-tree* if it is a connected $\mathbb{D}$-forward-forest with a unique root. We omit the set $\mathbb{D}$ and the adjective "forward" in the naming whenever it is known from the context or unimportant. An interpretation is *forward-tree-shaped* if it is a $\mathbb{D}$-forward-tree for some $\mathbb{D}$.

When working with forests it is convenient to employ the tailored terminology, borrowed from graph theory. Given a $\mathbb{D}$-forward-forest $\mathcal{I}$ we define an ordering $(\Delta^{\mathcal{I}}, \preceq)$ on it in such a way that $\mathrm{d} \preceq \mathrm{e}$ holds iff d is a prefix of e and use the following naming scheme:

- If $\mathrm{d} \prec \mathrm{e}$ holds (*i.e.* $\mathrm{d} \preceq \mathrm{e}$ but $\mathrm{d} \neq \mathrm{e}$) then d is an *ancestor* of e (e is a *descendant* of d).

- If $\mathrm{d}_1 \preceq \mathrm{d}_2$ but there is no element e such that $\mathrm{d}_1 \prec \mathrm{e} \prec \mathrm{d}_2$ we call $\mathrm{d}_1$ a *parent* of $\mathrm{d}_2$ or, alternatively, say that $\mathrm{d}_2$ is a *child* of $\mathrm{d}_1$. Note that it implies that there exists a value $\mathrm{c} \in \mathbb{D}$ such that $\mathrm{d}_2 = \mathrm{d}_1\mathrm{c}$.
- A *branch* in $\mathcal{I}$ is a (possibly infinite) sequence of elements $\mathrm{d}_1, \mathrm{d}_2, \ldots$ such that for any index $i \geq 0$ if an element $\mathrm{d}_{i+1}$ exists, then $\mathrm{d}_{i+1}$ is a child of $\mathrm{d}_i$.
- The $\prec$-maximal elements are called *leaves*.
- Given $\mathrm{d} \in \Delta^{\mathcal{I}}$ we denote the set of its children and its descendants, respectively, with $\mathsf{Chlds}_{\mathcal{I}}(\mathrm{d})$ and $\mathsf{Desc}_{\mathcal{I}}(\mathrm{d})$. We also define the *subtree* rooted at d, denoted: $\mathcal{I}^{[\mathrm{d}\preceq]}$, *i.e.* the restriction of $\mathcal{I}$ to the set $\{\mathrm{d}\} \cup \mathsf{Desc}_{\mathcal{I}}(\mathrm{d})$.

We conclude by lifting the notion of "being a forest" to models of knowledge bases. Take a set of individual names $\mathsf{N} \subseteq \mathbf{N_I}$. We say that a forward forest $\mathcal{I}$ is $\mathsf{N}$-*rooted* whenever:

- for all names $\mathsf{a} \in \mathsf{N}$ we have that $\mathsf{a}^{\mathcal{I}}$ is defined and it is a root of $\mathcal{I}$, and
- for each root $\mathrm{d} \in \Delta^{\mathcal{I}}$ there is a name $\mathsf{a} \in \mathsf{N}$ satisfying $\mathrm{d} = \mathsf{a}^{\mathcal{I}}$.

For convenience, we refer to forward trees as $\emptyset$-rooted forests. A *forward forest model* of a KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ is an $\mathsf{ind}(\mathcal{A})$-rooted forest satisfying $\mathcal{K}$.

### 2.4 Morphisms

Let $\mathcal{I}, \mathcal{J}$ be structures and let $\mathsf{N} \subseteq \mathbf{N_I}$. An $\mathsf{N}$-*homomorphism* $\mathfrak{f} : \mathcal{I} \to \mathcal{J}$ is a function that:

- maps $\Delta^{\mathcal{I}}$ to $\Delta^{\mathcal{J}}$,
- preserves individual names from $\mathsf{N}$, *i.e.* for all $\mathsf{a} \in \mathsf{N}$ if $\mathsf{a}^{\mathcal{I}}$ is defined then $\mathsf{a}^{\mathcal{J}} = \mathfrak{f}(\mathsf{a}^{\mathcal{I}})$,
- preserves atomic concepts, *i.e.* $\mathrm{d} \in \mathrm{A}^{\mathcal{I}}$ implies $\mathfrak{f}(\mathrm{d}) \in \mathrm{A}^{\mathcal{J}}$ for all $\mathrm{A} \in \mathbf{N_C}$,
- and preserves atomic roles, *i.e.* $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$ implies $(\mathfrak{f}(\mathrm{d}), \mathfrak{f}(\mathrm{e})) \in r^{\mathcal{J}}$ for all $r \in \mathbf{N_R}$.

An $\mathsf{N}$-*isomorphism* $\mathfrak{f} : \mathcal{I} \to \mathcal{J}$ is a bijection such that $\mathfrak{f}$ and $\mathfrak{f}^{-1}$ are $\mathsf{N}$-homomorphisms. We write $\mathcal{I} \lhd_{\mathsf{N}} \mathcal{J}$ to indicate that there is an $\mathsf{N}$-homomorphism from $\mathcal{I}$ to $\mathcal{J}$. In this case $\mathcal{I}$ is said to be $\mathsf{N}$-homomorphically mapped to $\mathcal{J}$. Structures $\mathcal{I}, \mathcal{J}$ are $\mathsf{N}$-*homomorphically equivalent*, written: $\mathcal{I} \rightleftarrows_{\mathsf{N}} \mathcal{J}$, if $\mathcal{I} \lhd_{\mathsf{N}} \mathcal{J}$ and $\mathcal{J} \lhd_{\mathsf{N}} \mathcal{I}$ hold. Finally, $\mathcal{I}$ and $\mathcal{J}$ are $\mathsf{N}$-isomorphic, written: $\mathcal{I} \cong_{\mathsf{N}} \mathcal{J}$, if there exists an $\mathsf{N}$-isomorphism between them. We often use the term homomorphism (resp. isomorphism) rather than $\emptyset$-homomorphism (resp. $\emptyset$-isomorphism).

Note that the composition of $\mathsf{N}$-homomorphisms is also an $\mathsf{N}$-homomorphism. Similarly, the composition of $\mathsf{N}$-isomorphisms is also an $\mathsf{N}$-isomorphism.

### 2.5 Queries

Queries employ *variables* from a countably infinite set $\mathbf{N_V}$. A *conjunctive query* (CQ) is a conjunction of *atoms* of the form $r(x, y)$ or $\mathrm{A}(z)$, where $r$ is a role name, A is a concept name and $x, y, z$ are variables. More expressive query languages are also considered: a *union of conjunctive queries* (UCQ) is a disjunction of CQs and a *positive existential query* (PEQ) is a positive boolean combination of CQs.[7] Note that any PEQ can be converted to a UCQ of (possibly) exponential size by turning it into disjunctive normal form.

7. PEQs are generated with the following grammar: $q ::= \mathrm{A}(x) \mid r(x, y) \mid q \wedge q \mid q \vee q$.

Let $q$ be a PEQ and let $\mathcal{I}$ be a structure. The set of variables appearing in $q$ is denoted with $\mathrm{Var}(q)$ and the number of atoms of $q$ (*i.e.* the size of $q$) is denoted with $|q|$. The fact that $r(x, y)$ appears in $q$ is indicated with $r(x, y) \in q$. Whenever some subset $V \subseteq \mathrm{Var}(q)$ is given, let $q{\restriction_V}$ denote the sub-query of $q$ where all the atoms containing any variable outside $V$ are removed. Let $\pi : \mathrm{Var}(q) \to \Delta^{\mathcal{I}}$ be a *variable assignment*. We write $\mathcal{I} \models_\pi r(x, y)$ if $(\pi(x), \pi(y)) \in r^{\mathcal{I}}$ and $\mathcal{I} \models_\pi \mathrm{A}(z)$ if $\pi(z) \in \mathrm{A}^{\mathcal{I}}$. Similarly, we write $\mathcal{I} \models_\pi q_1 \wedge q_2$ (resp. $\mathcal{I} \models_\pi q_1 \vee q_2$) iff $\mathcal{I} \models_\pi q_1$ and (resp. or) $\mathcal{I} \models_\pi q_2$, for queries $q_1, q_2$. We say that $\pi$ is a *match* for $\mathcal{I}$ and $q$ if $\mathcal{I} \models_\pi q$ holds and that $\mathcal{I}$ *satisfies* $q$ (denoted with: $\mathcal{I} \models q$) whenever $\mathcal{I} \models_\pi q$ for some match $\pi$. The definitions are lifted to knowledge bases: $q$ is *(finitely) entailed* by a knowledge base $\mathcal{K}$ (written: $\mathcal{K} \models_{(\mathrm{fin})} q$) if every (finite) model $\mathcal{I}$ of $\mathcal{K}$ satisfies $q$. We stress that the entailment relations $\models$ and $\models_{\mathrm{fin}}$ may not coincide (this is the case for the description logic $\mathcal{SQ}$ (Gutiérrez-Basulto, Ibáñez-García, & Jung, 2018, Example 3)). When $\mathcal{I} \models \mathcal{K}$ but $\mathcal{I} \not\models q$, we call $\mathcal{I}$ a *countermodel* for $\mathcal{K}$ and $q$. Note that $q$ is (finitely) entailed by $\mathcal{K}$ if there is no (finite) countermodel for $\mathcal{K}$ and $q$.

Observe that a *CQ* $q$ can be seen as a structure $\mathcal{I}_q = (\mathrm{Var}(q), \cdot^{\mathcal{I}_q})$, having the interpretation of roles and concepts fixed as $\mathrm{A}^{\mathcal{I}_q} = \{x \mid \mathrm{A}(x) \in q\}$ and $r^{\mathcal{I}_q} = \{(x, y) \mid r(x, y) \in q\}$ for all $\mathrm{A} \in \mathbf{N_C}$ and $r \in \mathbf{N_R}$ and with $\mathtt{a}^{\mathcal{I}_q}$ undefined for all $\mathtt{a} \in \mathbf{N_I}$. Hence, any match $\pi$ for $\mathcal{I}$ and CQ $q$ can be seen as an $\mathbf{N_I}$-homomorphism from $\mathcal{I}_q$ to $\mathcal{I}$. We say that a CQ $q$ is *forward-tree-shaped* whenever $\mathcal{I}_q$ is forward-tree-shaped.

For the class of *path-shaped conjunctive queries*, namely conjunctive queries whose query structure looks like a path, we often employ an alternative *path syntax* for conciseness. Thus, by a path-shaped conjunctive query we understand an expression of the form

$$(\mathrm{A}_0?; r_1; \mathrm{A}_1?; r_2; \mathrm{A}_2?; \ldots; \mathrm{A}_{n-1}?; r_n; \mathrm{A}_n?)(x_0, x_n)$$

with all $r_i \in \mathbf{N_R}$ and $\mathrm{A}_i \in \mathbf{N_C} \cup \{\top\}$, serving as a shorthand for

$$\bigwedge_{i=0}^{n} \mathrm{A}_i(x_i) \wedge \bigwedge_{i=1}^{n} r_i(x_{i-1}, x_i).$$

Whenever $\mathrm{A}_i$ happens to be $\top$, it will be removed from the expression; this does not create ambiguities. We stress that the alternative syntax for path-shaped CQs is just syntactic sugar and our queries should not be mistaken *e.g.* for regular path queries (RPQs).

Let us conclude the section by discussing the differences between our definitions of queries and the ones that are present in the literature. First, our queries are always assumed to be Boolean, *i.e.* we do not allow for answer variables. This assumption is done (Glimm, Lutz, Horrocks, & Sattler, 2008, p. 164) w.l.o.g. as answer variables can be simulated with quantified variables and additional concept names. Second, individual names are not present in atoms in queries. This is again w.l.o.g. as one can proceed for any knowledge-base $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ and any PEQ $q$ as follows. Take any individual name $\mathtt{a}$ appearing in query $q$ and proceed as follows: (i) introduce a fresh variable $x_{\mathtt{a}}$ and fresh concept name $\mathrm{A}_{\mathtt{a}}$, (ii) replace each atom $\alpha$ in $q$ involving $\mathtt{a}$ by $\alpha \wedge \mathrm{A}_{\mathtt{a}}(x_{\mathtt{a}})$, (iii) replace every occurrence of the individual name $\mathtt{a}$ in $q$ by $x_{\mathtt{a}}$, and (iv) append $\mathrm{A}_{\mathtt{a}}(\mathtt{a})$ to the ABox $\mathcal{A}$. Let $q'$, $\mathcal{K}'$ be the resulting query and the resulting knowledge base. It is not difficult to show that $\mathcal{K} \models_{(\mathrm{fin})} q$ if and only if $\mathcal{K}' \models_{(\mathrm{fin})} q'$.

### 2.6 Decision Problems

For a given description logic $\mathcal{DL}$ we consider the classical decision problems, namely the (finite) *satisfiability problem* and the (finite) CQ/UCQ/PEQ *entailment problem*. The former asks if an input knowledge base has a (finite) model, while in the latter asks if an input CQ/UCQ/PEQ is (finitely) entailed by an input knowledge base. Here we mention a few results on $\mathcal{ALC}$ and sister logics. It is well-known that $\mathcal{ALC}$ has the *finite model property* (Grädel, 1999, Thm. 3.10), *i.e.* the satisfiability and the finite satisfiability problems coincide. Moreover, $\mathcal{ALC}$ is *finitely controllable* (Bárány, Gottlob, & Otto, 2014, Thm. 1.2) that is, any UCQ is entailed by an $\mathcal{ALC}$ knowledge base iff it is finitely entailed. These two results rely on the fact that $\mathcal{ALC}$ can be encoded (Baader et al., 2017, Ch. 2.6.1) in the so-called *guarded fragment of first-order logic* $\mathsf{GF}$ (Andréka, Németi, & van Benthem, 1998). Regarding the complexity results, the satisfiability problem (De Giacomo & Lenzerini, 1996, Thm. 6) and the CQ-entailment problem for $\mathcal{ALC}$ (Ortiz, Simkus, & Eiter, 2008, Thm. 6) (and even $\mathcal{ALCHQ}$ (Lutz, 2008b, Thm. 1)) are ExpTime-complete, while the PEQ-entailment problem for $\mathcal{ALC}$ was shown to be 2ExpTime-hard (Ortiz & Simkus, 2014, Thm. 1). The 2ExpTime upper bound can be obtained even for very expressive extensions of $\mathcal{ALC}$ and regular queries extending PEQs (Calvanese, Eiter, & Ortiz, 2014, Thm. 5.23). The UCQ entailment problem for $\mathcal{ALCH}$ is known to be ExpTime-complete (Ortiz, 2010, Thm. 6.5.1), while the exact complexity of UCQ-querying for many logics, including $\mathcal{ALCQ}$, is still unknown. The absence of such results is even more intriguing in the light of the existing 2ExpTime-hardness proofs of CQ entailment for $\mathcal{ALCO}$ (Ngo et al., 2016, Thm. 9) and $\mathcal{ALCI}$ (Lutz, 2007, Thm. 2), *i.e.* the extensions of $\mathcal{ALC}$ with *nominals* or *inverses of roles*.

### 2.7 Alternating Turing Machines

We next fix the notation of alternating Turing machines over a binary alphabet $\{\mathtt{0},\mathtt{1}\}$ working in exponential space (simply: ATMs). As a convention, when speaking about ATMs, their configurations and tape contents, we employ the typewriter font. An ATM is a tuple $\mathcal{M} := (\mathtt{N}, \mathtt{Q}, \mathtt{Q}_\exists, \mathtt{s}_I, \mathtt{s}_A, \mathtt{s}_R, \mathtt{T})$, where $\mathtt{Q}$ is a finite set of *states* (usually denoted with $\mathtt{s}$); $\mathtt{Q}_\exists \subseteq \mathtt{Q}$ is a set of *existential* states; $\mathtt{s}_I, \mathtt{s}_A, \mathtt{s}_R \in \mathtt{Q}$ are, respectively, pairwise different *initial*, *accepting*, and *rejecting* states; we assume that $\mathtt{s}_I \in (\mathtt{Q} \setminus \mathtt{Q}_\exists)$; $\mathtt{T} \subseteq (\mathtt{Q} \times \{\mathtt{0},\mathtt{1}\}) \times (\{\mathtt{0},\mathtt{1}\} \times \mathtt{Q} \times \{-1,+1\})$ is the *transition relation*; and the natural number $\mathtt{N}$ (encoded in unary) is a parameter governing the size of the working tape. We call the states from $\mathtt{Q}_\forall := \mathtt{Q} \setminus \mathtt{Q}_\exists$ *universal*. The size of $\mathcal{M}$, denoted with $|\mathcal{M}|$, is defined as $\mathtt{N} + |\mathtt{Q}| + |\mathtt{Q}_\exists| + 3 + |\mathtt{T}|$.

A *configuration* of $\mathcal{M}$ is a word $\mathtt{wsw'} \in \{\mathtt{0},\mathtt{1}\}^*\mathtt{Q}\{\mathtt{0},\mathtt{1}\}^*$ with $|\mathtt{ww'}| = 2^\mathtt{N}$. We call $\mathtt{wsw'}$ (i) existential (resp. universal) if $\mathtt{s}$ is existential (resp. universal), (ii) final if $\mathtt{s}$ is either $\mathtt{s}_A$ or $\mathtt{s}_R$ (iii) non-final if it is not final (iv) accepting if $\mathtt{s} = \mathtt{s}_A$. Successor configurations are defined in terms of the transition relation $\mathtt{T}$. For $\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d} \in \{\mathtt{0},\mathtt{1}\}$ and $\mathtt{v},\mathtt{v'},\mathtt{w},\mathtt{w'} \in \{\mathtt{0},\mathtt{1}\}^*$ with $|\mathtt{v}| = |\mathtt{w}|$, we let $\mathtt{wbs'w'}$ be a *quasi-successor* configuration of $\mathtt{vsav'}$ whenever $(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s'},+1) \in \mathtt{T}$, and we let $\mathtt{ws'dbw'}$ be a quasi-successor configuration of $\mathtt{vcsav'}$ whenever $(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s'},-1) \in \mathtt{T}$. If additionally $\mathtt{w} = \mathtt{v}$, $\mathtt{w'} = \mathtt{v'}$, and $\mathtt{c} = \mathtt{d}$ hold we speak of *successor* configurations.[8]

W.l.o.g we make the following additional assumptions about $\mathcal{M}$: First, for each non-final (*i.e.* non-accepting and non-rejecting) state $\mathtt{s}$ and every letter $\mathtt{a} \in \{\mathtt{0},\mathtt{1}\}$ the set $\mathtt{T}(\mathtt{s},\mathtt{a}) :=$

8. In words, this corresponds to the common definition of successor configurations, while for quasi-successor configurations, untouched tape cells may change arbitrarily during the transition.

$\{(\mathtt{s}, \mathtt{a}, \mathtt{b}, \mathtt{s}', d) \in \mathtt{T}\}$ contains exactly two elements, denoted $\mathtt{T}_1(\mathtt{s}, \mathtt{a})$ and $\mathtt{T}_2(\mathtt{s}, \mathtt{a})$. Hence, every configuration has exactly two successor configurations. Second, for any $(\mathtt{s}, \mathtt{a}, \mathtt{b}, \mathtt{s}', d) \in \mathtt{T}$, if $\mathtt{s}$ is existential then $\mathtt{s}'$ is universal and vice versa. Third, the machine reaches a final state no later than after $2^{2^N}$ steps (for configuration sequences). Fourth and last, $\mathcal{M}$ never attempts to move left (resp. right) on the left-most (resp. right-most) tape cell.

A *run* of $\mathcal{M}$ is a finite tree, with nodes labelled by configurations of $\mathcal{M}$, that satisfies all the conditions below:

- the root is labelled with the initial configuration $\mathtt{s}_I\mathtt{0}^{2^N}$,
- each node labelled with a non-final existential configuration $\mathtt{wsw}'$ has a single child node which is labelled with one of the successor configurations of $\mathtt{wsw}'$,
- each node labelled with a non-final universal configuration $\mathtt{wsw}'$ has two child nodes which are labelled with the two successor configurations (wrt. $\mathtt{T}_1$ and $\mathtt{T}_2$) of $\mathtt{wsw}'$,
- no node labelled with a final configuration has successors.

*Quasi-runs* of $\mathcal{M}$ are defined analogously by replacing the notions of successors with quasi-successors. Note that every run is also a quasi-run but not vice versa.

An ATM $\mathcal{M}$ is (quasi-)*accepting* if it has an *accepting* (quasi)-run, *i.e.* one whose all leaves are labelled by accepting configurations. By results of Chandra et al. (1981, Corollary 3.6) the problem of checking if a given ATM is accepting is 2ExpTime-hard.

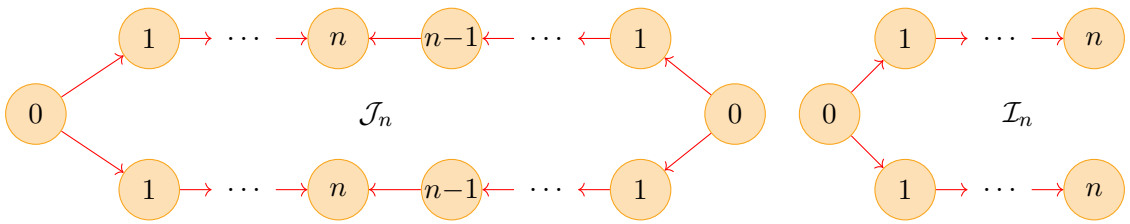
## 3. Query Entailment in Locally-Forward Description Logics

In this section, we provide a worst-case optimal algorithm for solving (U)CQ entailment problem for the classes $\mathscr{C}_{\mathsf{lf}}$ and $\mathscr{C}_{\mathsf{flf}}$ of (finitely) locally-forward abstract DLs. Let us start by defining what locally-forward abstract DLs actually are.

**Definition 3.1.** *For $n \in \mathbb{N}$ and a set of names $\mathsf{N} \subseteq \mathbf{N_I}$, we say that an interpretation $\mathcal{I}$ is $(n, \mathsf{N})$-*locally-forward-forest-like *(short: $(n, \mathsf{N})$-*lff-like*) iff every $n$-neighbourhood $\mathcal{J}$ in $\mathcal{I}$ is $(\mathsf{ind}(\mathcal{J}) \cap \mathsf{N})$-homomorphically equivalent to some $(\mathsf{ind}(\mathcal{J}) \cap \mathsf{N})$-rooted forward forest.*

Informally, any sufficiently small (of radius $n$) neighbourhood of $(n, \mathsf{N})$-lff-like structures resembles a forward tree or a forward forest. We stress here that lff-like structures may contain undirected cycles composed of anonymous elements:

**Remark 3.2.** *Consider a forward tree $\mathcal{I}_n$ (right), and a structure $\mathcal{J}_n$ (left) that is composed of $\mathcal{I}_n$ and its mirrored image glued together, depicted below. Clearly $\mathcal{I}_n$ and $\mathcal{J}_n$ are homomorphically-equivalent. On the other hand $\mathcal{I}_n$ is acyclic and $\mathcal{J}_n$ contains an undirected cycle of size $> n$. Thus lff-like structures may contain arbitrarily large undirected cycles.*



Locally-forward-forest-like structures are next employed as "coverings" of interpretations. The property below is inspired by the quasi-forest homomorphism-covers by Bourhis, Krötzsch, and Rudolph (2014, p. 8).

**Definition 3.3** (coverable by lffs)**.** *Let $\mathcal{DL}$ be an abstract DL and let $\mathcal{K}$ be an $\mathcal{DL}$-KB. Call $\mathcal{K}$ (finitely)* coverable by locally-forward-forest-like structures *(short: lff-coverable) iff for any (finite) model $\mathcal{I} \models \mathcal{K}$ and for every $n \in \mathbb{N}$ there is a (finite) $(n, \mathsf{ind}(\mathcal{K}))$*-lff-like *model $\mathcal{J} \models \mathcal{K}$ that* covers *$\mathcal{I}$,* i.e. *any $n$-neighbourhood of $\mathcal{J}$ can be $\mathsf{ind}(\mathcal{K})$-homomorphically-mapped to $\mathcal{I}$.*

Finally we employ coverings and lff-like interpretations to define locally-forward DLs.

**Definition 3.4.** *An abstract DL $\mathcal{DL}$ is said to be (finitely)* locally-forward *iff all $\mathcal{DL}$-KBs are (finitely) lff-coverable. We write $\mathscr{C}_{\mathsf{lf}}$ and $\mathscr{C}_{\mathsf{flf}}$ to denote the classes of locally-forward and finitely locally-forward abstract DLs.*

The above-defined notion of coverability is just a technicality, which is not very informative when dealing with logics from $\mathscr{C}_{\mathsf{lf}} \cup \mathscr{C}_{\mathsf{flf}}$. What actually matters is the following property, relying on the fact that UCQs are closed under homomorphisms.

**Property 3.5.** *For any (finitely) locally-forward $\mathcal{DL}$, any $\mathcal{DL}$-KB $\mathcal{K}$ and any UCQ $q$ of the form $\bigvee_{i=1}^{m} q_i$ with $\mathcal{K} \not\models q$, there is a (finite) $(|q|, \mathsf{ind}(\mathcal{K}))$-lff-like countermodel for $\mathcal{K}$ and $q$.*

*Proof sketch.* Take a countermodel $\mathcal{I}$ for $\mathcal{K}$ and $q$, and its promised locally-forward cover $\mathcal{J}$. Towards contradiction assume $\mathcal{J} \models q_i$ for some $q_i$. By coverability, any $|q|$-neighbourhood of $\mathcal{J}$ can be homomorphically mapped to $\mathcal{I}$, thus transferring a match of $q_i$ from $\mathcal{J}$ to $\mathcal{I}$. □

One can provide multiple examples of locally-forward abstract DLs. To do so, observe that any logic $\mathcal{DL}$ extending $\mathcal{ALC}^{\cap}$ and having the forward-forest-countermodel property is immediately locally-forward. More precisely, for any $\mathcal{DL}$-KB $\mathcal{K}$, a guaranteed forward-forest countermodel $\mathcal{I}$ for $\mathcal{K}$ and a (U)CQ of size $n$ is $(n, \mathsf{ind}(\mathcal{K}))$-lff-like. The following statement is a straightforward consequence of existing work.

**Corollary 3.6.** *Any extension of $\mathcal{ALC}^{\cap}$ contained in $\mu\mathcal{ALCSCC}$ has an* ExpTime*-complete knowledge base satisfiability problem and belongs to $\mathscr{C}_{\mathsf{lf}}$.*

*Proof sketch.* The first part of the statement follows by a minor adaptation of a proof by Bonatti et al. (2008, Theorem 3.3) (a self-contained proof can be obtained by employing sufficient conditions presented in Section 4). The second claim follows from a work of Kupke et al. (2022), presented under a very general, co-algebraic setting. □

There are also logics for which only the finite model semantics makes sense. One such example is $\mathcal{ALCSCC}$ extended with ERCBoxes (Baader et al., 2020), *i.e.* positive boolean combinations of linear inequalities over the domain (such constraints become trivial outside the realm of finite models). In Section 4.3, we prove the membership of $\mathcal{ALCSCC}$+ERCBoxes in $\mathscr{C}_{\mathsf{flf}}$, which will allow us to deduce precise upper bounds on the complexity of its querying.

There is a plethora of logics that are not (finitely) locally-forward, for instance $\mathcal{ALC}^{\cap}$ extended with nominals, inverses, the Self operator, or transitivity. Although all these logics enjoy "forest-like" models, they are not locally-forward due to the presence of "backlinks" (edges going back from an element to a nominal), bidirectional edges, self-loops or edges linking an element to its distant descendant. We leave the verification of this fact to the reader.

### 3.1 An Informal Explanation of Lutz's Spoiler Technique

We start by giving a rather informal explanation of Lutz's spoiler technique, dedicated to the readers that are not familiar with the original work of Lutz on querying $\mathcal{ALCHQ}$ (2008b, Sec. 3).[9] Most of the forthcoming notions are inspired by those from the work of Lutz (2008b) and actually we aimed at reusing as much material from his work as possible. However, many of our statements require separate proofs in order to make them logic-independent and to adjust the proof to locally-forest-like structures.

Recall that our goal is to decide, given a finitely lff-coverable $\mathcal{DL}$-KB $\mathcal{K}$ and a conjunctive query $q$, whether $\mathcal{K} \models_{(\text{fin})} q$ holds, which boils down to checking if there is a (finite or arbitrary, depending on the problem) countermodel for $\mathcal{K}$ and $q$. Due to Property 3.5 we can restrict our attention to $(n, \mathsf{ind}(\mathcal{K}))$-lff-like interpretations. An important observation is that a match $\pi$ of $q$ over an $(|q|, \mathsf{ind}(\mathcal{K}))$-lff-like $\mathcal{I}$ induces a very specific partition of $\mathrm{Var}(q)$, namely $\pi$ divides the variables of $q$ into three disjoint categories: (i) the variables mapped to the $\mathsf{N}$-named elements of $\mathcal{I}$, (ii) the variables forming a forward subtree "dangling" from one of the $\mathsf{N}$-named elements of $\mathcal{I}$ and (iii) the variables forming forward-trees that lie "far" from $\mathsf{N}$-named elements. The notion of a *splitting* abstractly describes such a partition, independently of the choice of $\pi$ and $\mathcal{I}$. The existence of a splitting *compatible* with a $(|q|, \mathsf{ind}(\mathcal{K}))$-lff-like $\mathcal{I}$ implies that $\mathcal{I} \models q$ holds and vice versa. Hence, to show $\mathcal{K} \not\models_{(\text{fin})} q$, it suffices to find a (finite) $(|q|, \mathsf{ind}(\mathcal{K}))$-lff-like model $\mathcal{I}^{\lightning}$ of $\mathcal{K}$ such that no splitting is compatible with it, or, in other words, that $\mathcal{I}^{\lightning}$ *spoils* all the splittings. Next, for a splitting $\Pi_q$ of $q$ we design an $\mathcal{DL}$-KB $\mathcal{K}^{\lightning}_{\Pi_q}$, called a *spoiler* for $\Pi_q$ with the intended meaning that every $(|q|, \mathsf{ind}(\mathcal{K}))$-locally-forward-forest-like model of $\mathcal{K} \cup \mathcal{K}^{\lightning}_{\Pi_q}$ *spoils* its compatibility with $\Pi_q$. The construction of spoilers employs, among other ingredients, the well-known *rolling-up technique* (Horrocks & Tessaris, 2000, Sec. 4) that is used to detect forward-tree-shaped query matches from points (ii)–(iii) above (the name of the technique comes from the fact that we traverse an input forward-tree in a bottom-up manner and gradually "rolling-up" its forward subtrees into predicates, until the root is reached). This is the only reason why we require that $\mathcal{DL}$ polynomially encodes $\mathcal{ALC}^{\cap}$ concepts. Having the splittings defined, we observe that (finite) $(|q|, \mathsf{ind}(\mathcal{K}))$-lff-like models of $\mathcal{K} \cup \bigcup_{\Pi_q} \mathcal{K}^{\lightning}_{\Pi_q}$ are (finite) countermodels for $\mathcal{K}$ and $q$.

This yields decidability, but with a suboptimal complexity when the (finite) satisfiability problem for $\mathcal{DL}$ is ExpTime-complete. To get the optimal (exponential) upper bound in such a case, we parallelise the construction of $\bigcup_{\Pi_q} \mathcal{K}^{\lightning}_{\Pi_q}$. This means, intuitively, that the KB $\bigcup_{\Pi_q} \mathcal{K}^{\lightning}_{\Pi_q}$ is divided into exponentially many chunks called *super-spoilers* $\mathcal{K}^{\lightning\star}_q$ with the meaning that $\mathcal{K} \not\models_{(\text{fin})} q$ iff $\mathcal{K} \cup \mathcal{K}^{\lightning\star}_q$ has a (finite) $(|q|, \mathsf{ind}(\mathcal{K}))$-lff-like model for some super-spoiler $\mathcal{K}^{\lightning\star}_q$. We then show that each super-spoiler is of polynomial size and the set of super-spoilers can be enumerated in exponential time. This gives us a Turing reduction from the (finite) query entailment problem to exponentially many (finite) satisfiability checks of polynomial-size $\mathcal{DL}$-KBs, which yields an optimal complexity.

9. Lutz works with $\mathcal{SHQ}$, an extension of $\mathcal{ALCHQ}$ with transitive roles, but he does not allow for transitive roles in queries. This is crucial since their presence makes the CQ entailment problem exponentially harder (Eiter et al., 2009, Thm. 1). Hence, Lutz's work is more about querying $\mathcal{ALCHQ}$ than $\mathcal{SHQ}$.

### 3.2 Step I: Rolling-up Forward-Tree-Shaped Queries

We next recall the well-known rolling-up technique (Horrocks & Tessaris, 2000, Sec. 4) of transforming forward-tree-shaped queries into $\mathcal{ALC}^{\cap}$-concepts. Our goal is to construct, for every $x \in \mathrm{Var}(q)$, a concept $\mathrm{Subt}_q^x$ stating that $\mathrm{d} \in (\mathrm{Subt}_q^x)^{\mathcal{I}}$ holds whenever the subtree of $\mathcal{I}_q$ rooted at the variable $x$ can be mapped below d in $\mathcal{I}$ (made precise in Lemma 3.8). A formal, inductive definition is given next. The main idea behind the definition is to traverse the input tree in a bottom-up manner, describing its shape with $\mathcal{ALC}^{\cap}$ concepts, and gradually "rolling-up" the input forward-tree into smaller chunks until the root is reached.

**Definition 3.7.** *For a forward-tree-shaped CQ $q$ and any of its variables $v \in \mathrm{Var}(q)$ we define an $\mathcal{ALC}^{\cap}$-concept* $\mathrm{Subt}_q^v$ *as:*

$$\mathrm{Subt}_q^v \coloneqq \bigsqcap_{\mathrm{A}(v) \in q} \mathrm{A} \;\sqcap \bigsqcap_{u \in \mathsf{Chlds}(v)} \exists \left( \bigcap_{r(v,u) \in q} r \right) . \mathrm{Subt}_q^u,$$

*where the empty conjunction equals $\top$. We use* $\mathrm{Match}_q$ *as an abbreviation of* $\mathrm{Subt}_q^{v_r}$ *with $v_r$ being the root of $\mathcal{I}_q$.*

From the presented construction we can easily estimate the size (*i.e.* the number of sub-concepts) of $\mathrm{Match}_q$. Note that the size of $\mathrm{Match}_q$ is linear in $|q|$ since every query atom contributes to exactly one sub-concept of $\mathrm{Match}_q$. The following lemma is folklore:

**Lemma 3.8.** *For any interpretation $\mathcal{I}$, any forward-tree-shaped conjunctive query $q$ and any of its variables $v \in \mathrm{Var}(q)$, the following equivalence holds:* $\mathrm{d} \in (\mathrm{Subt}_q^v)^{\mathcal{I}}$ *iff there exists a homomorphism $\mathfrak{h} : \mathcal{I}_q^{[v \preceq]} \to \mathcal{I}$ with $\mathfrak{h}(v) = \mathrm{d}$, where $\mathcal{I}_q^{[v \preceq]}$ denotes the subtree of $\mathcal{I}_q$ rooted at $v$.*

*Proof sketch.* Routine induction over $(\mathrm{Var}(q), \preceq)$ with the inductive assumption given above.

By unravelling the definition of $\mathrm{Match}_q$ and by applying Lemma 3.8 for the root variable of $q$, as an immediate consequence we conclude:

**Corollary 3.9.** *For any interpretation $\mathcal{I}$ and a forward-tree-shaped conjunctive query $q$ we have that* $(\mathrm{Match}_q)^{\mathcal{I}} \neq \emptyset$ *iff there exists a homomorphism $\mathfrak{h} : \mathcal{I}_q \to \mathcal{I}$.*

Informally, any element belonging to the interpretation of $\mathrm{Match}_q$ "detects" a match of the forward-tree-shaped $q$. Thus, one can employ Corollary 3.9 to design an algorithm for the entailment problem for forward-tree-shaped queries: for a given input query $q$ and a KB $\mathcal{K}$, it suffices to test whether $\mathcal{K} \cup \{\top \sqsubseteq \neg \mathrm{Match}_q\}$ has a model. Unfortunately, the presented method of detecting query matches works only for forward-tree-shaped queries. To detect matches of arbitrary CQs, we require a stronger mechanism, namely the notions of fork rewritings and splittings, which we will introduce in the following sections.

### 3.3 Step II: Fork Rewritings

A conjunctive query can induce many different query matches. In our case the target structures are lff-like, thus such matches are of a very specific form. For instance, every node inside a forward-tree has a unique parent. Thus, whenever we have a query match $\pi$ of

a query containing a "fork" $r(x,z) \wedge s(y,z)$, it clearly must be that the variables $x$ and $y$ are mapped via $\pi$ to the same element. We formalise this intuition with the forthcoming notion of fork rewritings (Lutz, 2008b, p. 4), that removes this kind of "redundancy" from queries.

**Definition 3.10.** *Let $q, q'$ be CQs. We say that $q'$ is obtained from $q$ by* fork elimination*, and denote this fact with $q \rightsquigarrow_{\mathsf{fe}} q'$, if $q'$ can be obtained from $q$ by selecting two atoms $r(y,x)$, $s(z,x)$ of $q$ (where $r$ and $s$ are not necessarily different) and identifying the variables $y, z$. We also say that $q'$ is a* fork rewriting *of $q$ if $q'$ is obtained from $q$ by applying fork elimination on $q$ possibly multiple times. When the fork elimination process is applied exhaustively on $q$ we say that the resulting query, denoted with $\mathsf{maxfr}(q)$, is the* maximal fork rewriting *of $q$.*

The proof of the following lemma is by Lutz (2008b, Appendix A).

**Lemma 3.11** (Lemma 1 by Lutz (2008b))**.** *For any conjunctive query $q$, there exists its (up to a variable renaming) unique maximal fork rewriting $\mathsf{maxfr}(q)$.*

To gain more intuitions on how the fork elimination works, consult the example below.

**Example 3.12.** *Consider a conjunctive query $q := r(x,y) \wedge r(x,z) \wedge s(v,y) \wedge r(v,z) \wedge \mathrm{A}(x) \wedge \mathrm{B}(y) \wedge \mathrm{C}(z) \wedge \mathrm{D}(v)$. By applying fork elimination for variables $x$ and $v$ we obtain the maximal fork rewriting of $q$,* i.e. *the conjunctive query $\mathsf{maxfr}(q) := r(xv,y) \wedge s(xv,y) \wedge r(xv,z) \wedge \mathrm{B}(y) \wedge \mathrm{A}(xv) \wedge \mathrm{D}(xv) \wedge \mathrm{C}(z)$, with $xv$ being a fresh variable.*


Figure 1: An example conjunctive query (left) and its maximal fork rewriting (right).

A rather immediate application of Definition 3.10 yields that an entailment of a fork rewriting of a query implies the entailment of the input query itself.

**Lemma 3.13.** *Let $q, q'$ be conjunctive queries, such that $q'$ is obtained from $q$ by fork rewriting, and let $\mathcal{I}$ be a structure. Then $\mathcal{I} \models q'$ implies $\mathcal{I} \models q$.*

*Proof sketch.* Assuming $\mathcal{I} \models q'$, there is derivation $q{=}q_n \rightsquigarrow_{\mathsf{fe}} q_{n-1} \rightsquigarrow_{\mathsf{fe}} \ldots \rightsquigarrow_{\mathsf{fe}} q_0{=}q'$. Reason inductively employing the fact that given a homomorphism $\mathfrak{h}_i : \mathcal{I}_{q_i} \to \mathcal{I}$ one can obtain a homomorphism $\mathfrak{h}_{i+1} : \mathcal{I}_{q_{i+1}} \to \mathcal{I}$ from $\mathfrak{h}_i$ simply by assigning to both $\mathfrak{h}_{i+1}(x)$ and $\mathfrak{h}_{i+1}(y)$ the value $\mathfrak{h}_i(x)$, where $x$ and $y$ were identified during the $i$-th fork elimination. □

### 3.4 Step III: Splittings

The next notion of splittings provides an abstraction of how a CQ $q$ matches a $(|q|, \mathsf{N})$-locally-forward-forest-like interpretation, while referring neither to a concrete interpretation nor to a concrete match. Intuitively, the role of splittings is to partition the variables $v$ of some fork rewriting $q$ of the input query, depending on the three possible scenarios:

- either $v$ is expected to be mapped to one of the $\mathsf{N}$-named elements,

- or $v$, together with some other variables, are expected to be mapped such that they form a subtree dangling from one of the $\mathsf{N}$-named elements,
- or $v$ is expected to be mapped somewhere "further down" inside the structure, disconnected from the $\mathsf{N}$-named elements.

**Definition 3.14.** *Let $\mathsf{N} \subseteq \mathbf{N_I}$ and let $q$ be a CQ. An* $\mathsf{N}$-splitting $\Pi_q^{\mathsf{N}}$ *of $q$ is a tuple*

$$\Pi_q^{\mathsf{N}} := (\mathtt{Roots}, \mathtt{name}, \mathtt{SubTree}_1, \mathtt{SubTree}_2, \dots, \mathtt{SubTree}_n, \mathtt{root\text{-}of}, \mathtt{Trees}) \,,$$

*where the sets $\mathtt{Roots}, \mathtt{SubTree}_1, \dots, \mathtt{SubTree}_n, \mathtt{Trees}$ induce a partition of $\mathrm{Var}(q)$, $\mathtt{name} : \mathtt{Roots} \to \mathsf{N}$ is a function naming the roots and $\mathtt{root\text{-}of} : \{1, 2, \dots, n\} \to \mathtt{Roots}$ assigns to each $\mathtt{SubTree}_i$ an element from $\mathtt{Roots}$. Moreover, to be an $\mathsf{N}$-splitting, $\Pi_q^{\mathsf{N}}$ has to satisfy all the conditions below:*

*(a) the query $q{\restriction_{\mathtt{Trees}}}$ is a conjunction of variable-disjoint forward-tree-shaped queries,*

*(b) the queries $q{\restriction_{\mathtt{SubTree}_i}}$ are forward-tree-shaped for all indices $i \in \{1, 2, \dots, n\}$,*

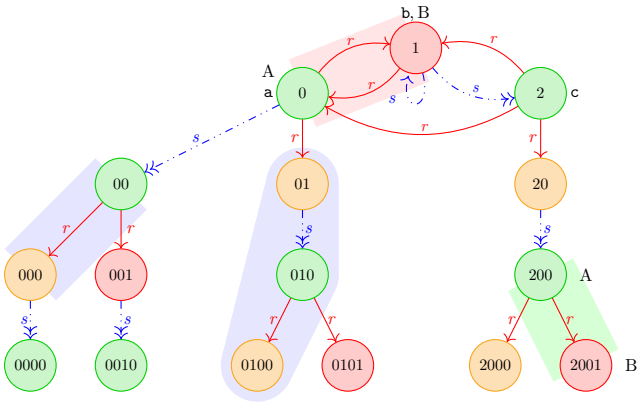
*(c) for any atom $r(x, y) \in q$ the variables $x, y$ either belong to the same set or there is an index $i \in \{1, 2, \dots, n\}$ such that $\mathtt{root\text{-}of}(i) = x \in \mathtt{Roots}$ and $y \in \mathtt{SubTree}_i$ is the root of $q{\restriction_{\mathtt{SubTree}_i}}$,*

*(d) For any index $i \in \{1, 2, \dots, n\}$ there is an atom $r(\mathtt{root\text{-}of}(i), x_i) \in q$ with $x_i$ being the root of $q{\restriction_{\mathtt{SubTree}_i}}$.*

It may help to think that a splitting represents an abstraction of an image of the query in a target structure, consisting of named roots, corresponding to the ABox part of the model, together with some of their subtrees and of some auxiliary forward-trees lying somewhere detached from the roots.

**Example 3.15.** *Consider an $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$-rooted forward forest $\mathcal{I}$ and a (non-tree-shaped) CQ $q$:*

$$q := (\mathrm{A}(x_0) \wedge r(x_0, x_1) \wedge r(x_1, x_0) \wedge \mathrm{B}(x_1)) \wedge (s(x_0, x_{00}) \wedge r(x_{00}, x_{000})) \\ \wedge (r(x_0, x_{01}) \wedge s(x_{01}, x_{010}) \wedge r(x_{010}, x_{0100})) \wedge (\mathrm{A}(x_{200}) \wedge r(x_{200}, x_{2001}) \wedge \mathrm{B}(x_{2001})) \,.$$




$$\mathtt{Roots} = \{x_0, x_1\}$$

$$\mathtt{SubTree}_1 = \{x_{00}, x_{000}\}$$

$$\mathtt{SubTree}_2 = \{x_{01}, x_{010}, x_{0100}\}$$

$$\mathtt{Trees} = \{x_{200}, x_{2001}\}$$

$$\mathtt{name}(x_0) = \mathtt{a}, \mathtt{name}(x_1) = \mathtt{b}$$

$$\mathtt{root\text{-}of}(1) = x_0, \mathtt{root\text{-}of}(2) = x_0$$

Figure 2: The left part of the picture depicts an example forest $\mathcal{I}$ that satisfies the query $q$. An example match of $q$, defined as $x_i \mapsto i$, is visualized by highlighted areas of the picture. The query itself is not depicted here. The right part of the pictures gives an example splitting $\Pi_q^{\{\mathsf{a},\mathsf{b},\mathsf{c}\}}$ of $q$, which corresponds to the presented match, and which is compatible with $\mathcal{I}$.

*Then a splitting* $\Pi_q^{\{a,b,c\}} := (\texttt{Roots}, \texttt{name}, \texttt{SubTree}_1, \texttt{SubTree}_2, \texttt{root-of}, \texttt{Trees})$ *defined in Figure 2 is compatible with* $\mathcal{I}$ *(consult Definition 3.16).*

We finish the section by showing that splittings indeed fulfil their purposes. In order to do it, we first introduce an immediate definition of *compatibility* of a splitting with a $(|q|, \mathsf{N})$-locally-forward-forest-like interpretation.

**Definition 3.16.** *Let* $\mathsf{N} \subseteq \mathbf{N_I}$*,* $q$ *be a CQ and* $\mathcal{I}$ *be a* $(|q|, \mathsf{N})$*-locally-forward-forest-like interpretation. We say that an* $\mathsf{N}$*-splitting* $\Pi_q^{\mathsf{N}}$ *of* $q$ *is* compatible *with* $\mathcal{I}$ *if all the conditions below are satisfied.*

*(A) for each connected component* $\hat{q}$ *of* $q{\restriction_{\texttt{Trees}}}$ *there is* $\mathrm{d} \in \Delta^{\mathcal{I}}$ *satisfying* $\mathrm{d} \in (\mathrm{Match}_{\hat{q}})^{\mathcal{I}}$*,*

*(B) for all atoms* $\mathrm{A}(x) \in q$ *with* $x \in \texttt{Roots}$ *we have* $(\texttt{name}(x))^{\mathcal{I}} \in \mathrm{A}^{\mathcal{I}}$*,*

*(C) for all atoms* $r(x, y) \in q$ *with* $x, y \in \texttt{Roots}$ *we have* $\left(\texttt{name}(x)^{\mathcal{I}}, \texttt{name}(y)^{\mathcal{I}}\right) \in r^{\mathcal{I}}$*,*

*(D) for all indices* $i \in \{1, 2, \ldots, n\}$ *the following property, for* $x_i$ *being the root of* $q{\restriction_{\texttt{SubTree}_i}}$*, is satisfied:*

$$\texttt{name}(\texttt{root-of}(i))^{\mathcal{I}} \in \left( \exists \left( \bigcap_{r(\texttt{root-of}(i), x_i) \in q} r \right) .\mathrm{Match}_{q{\restriction_{\texttt{SubTree}_i}}} \right)^{\mathcal{I}}.$$

The forthcoming lemmas link together all the notions presented in this section. Their proof is quite delicate and relies on an exhaustive case analysis: we give the main ideas below.

**Lemma 3.17.** *Let* $q$ *be a CQ,* $\mathsf{N} \subseteq \mathbf{N_I}$ *and* $\mathcal{I}$ *be a* $(|q|, \mathsf{N})$*-lff-like interpretation. Then* $\mathcal{I} \models q$ *if and only if there is a fork rewriting* $q'$ *of* $q$ *and an* $\mathsf{N}$*-splitting* $\Pi_{q'}^{\mathsf{N}}$ *of* $q'$*, such that* $\Pi_{q'}^{\mathsf{N}}$ *is compatible with* $\mathcal{I}$*.*

*Proof sketch (⇐).* By Lemma 3.13, it suffices to show $\mathcal{I} \models q'$. We define a homomorphism $\mathfrak{h} : \mathrm{Var}(q') \to \mathcal{I}$, witnessing $\mathcal{I} \models q'$, as follows:

- For every root variable $x \in \texttt{Roots}$ we put $\mathfrak{h}(x) := (\texttt{name}(x))^{\mathcal{I}}$.
- Fix an index $1 \leq i \leq n$. By Item (b) of Definition 3.14 we know that $q'{\restriction_{\texttt{SubTree}_i}}$ is forward-tree-shaped and let $x_i$ be its root. Moreover, by Item (D) of Definition 3.16 there exists an element $\mathrm{d}_i \in \Delta^{\mathcal{I}}$ satisfying:

$$\left(\texttt{name}(\texttt{root-of}(i))^{\mathcal{I}}, \mathrm{d}_i\right) \in \left( \bigcap_{r(\texttt{root-of}(i), x_i) \in q'} r^{\mathcal{I}} \right) \quad \text{and} \quad \mathrm{d}_i \in \left(\mathrm{Match}_{q'{\restriction_{\texttt{SubTree}_i}}}\right)^{\mathcal{I}}.$$

  From the forward-tree-shapedness of $q'{\restriction_{\texttt{SubTree}_i}}$ and Lemma 3.8 we conclude the existence of a homomorphism $\mathfrak{h}_i$ from $\mathcal{I}_{q'{\restriction_{\texttt{SubTree}_i}}}$ to $\mathcal{I}$ with $\mathfrak{h}_i(x_i) = \mathrm{d}_i$. Thus we can simply put $\mathfrak{h}(x) := \mathfrak{h}_i(x)$ for all $x \in \texttt{SubTree}_i$.
- Take any connected component $\hat{q}$ of $q'{\restriction_{\texttt{Trees}}}$, which by Item (a) of Definition 3.14 is forward-tree-shaped. From the compatibility of $\Pi_{q'}^{\mathsf{N}}$ with $\mathcal{I}$ and Item (A) of Definition 3.16 we know that there is an element $\mathrm{d} \in \Delta^{\mathcal{I}}$ satisfying $\mathrm{d} \in (\mathrm{Match}_{\hat{q}})^{\mathcal{I}}$. Invoking Corollary 3.9, we deduce that there exists a homomorphism $\mathfrak{h}_{\hat{q}} : \mathcal{I}_{\hat{q}} \to \mathcal{I}$. Finally, we put $\mathfrak{h}(x) := \mathfrak{h}_{\hat{q}}(x)$ for all $x \in \mathrm{Var}(\hat{q})$.

Showing that the definition of $\mathfrak{h}$ is correct relies on a routine case analysis, employing Definition 3.14 and Definition 3.16. □

*Proof sketch ($\Rightarrow$).* Let $\pi$ be a match witnessing $\mathcal{I} \models_\pi q$. As a preliminary step, we modify both $\pi$ and $q$ slightly, to make them "forest-like". An appropriate splitting is given afterwards.

The query $q'$ is constructed by exhaustive application of fork elimination on all "forks" $r(x, z), s(y, z)$ with $\pi(x) = \pi(y)$ (where $r, s$ are not necessarily different). Obviously $\mathcal{I} \models q'$ holds, witnessed by some match $\pi'$. Take any connected substructure $\mathcal{I}'$ of $\mathcal{I}$ induced by $\pi'$ and let $\mathsf{N}' \subseteq \mathsf{N}$ be names given to elements from $\mathcal{I}'$. Let $V$ be the set of all variables mapped to $\mathcal{I}'$. Depending on whether $\mathsf{N}'$ is empty or not, we invoke the assumption that $\mathcal{I}'$ is homomorphically equivalent to some forward-tree (resp. appropriately rooted forward forest) and modify the match $\pi'$ on $V$ so that $\pi'(V)$ is also a forward-tree (resp. forward-forest).

Finally, we define an $\mathsf{N}$-splitting

$$\Pi^{\mathsf{N}}_{q'} := (\texttt{Roots}, \texttt{name}, \texttt{SubTree}_1, \texttt{SubTree}_2, \ldots, \texttt{SubTree}_n, \texttt{root-of}, \texttt{Trees}),$$

where the definitions of its components are provided below.

- The set $\texttt{Roots}$ is composed of all variables $x \in \mathrm{Var}(q')$ for which $\pi'(x)$ is an $\mathsf{N}$-named element of $\mathcal{I}$. For all such variables $x$ we set $\texttt{name}(x) := \mathtt{a}$ for any corresponding $\mathtt{a} \in \mathsf{N}$.
- The sets $\texttt{SubTree}_i$, as their name suggests, are defined by taking subtrees connected to the roots. To simplify the definition, we say that a variable $x$ is *dangling from a root* if there exists a variable $x_r \in \texttt{Roots}$ and an atom $r(x_r, x)$ in $q'$. Let $D$ be the subset-maximal set of variables from $(\mathrm{Var}(q') \setminus \texttt{Roots})$ dangling from roots. Take $n := |D|$ and fix an ordering $x_1, x_2, \ldots, x_n$ on the elements from $D$. For any index $1 \leq i \leq n$ we define $\texttt{SubTree}_i$ as the set composed of $x_i$ and all variables reachable from $x_i$ via a directed path of positive length in the query structure $\mathcal{I}_{q'\restriction_{\mathrm{Var}(q')\setminus\texttt{Roots}}}$.
- We put $\texttt{root-of}(i) := x^i_r$, where $x^i_r$ is the root from which $x_i \in D$ is dangling.
- The set $\texttt{Trees}$ contains all other variables from $\mathrm{Var}(q')$.

Showing that $\Pi^{\mathsf{N}}_{q'}$ is indeed a splitting is quite delicate and relies on a careful analysis of multiple cases. We leave it for the appendix. □

Following Lutz, we say that a role conjunction $s_1 \cap \ldots \cap s_n$ *occurs* in a CQ $q$ if we can find two variables $v, v' \in \mathrm{Var}(q)$ such that $\{r \in \mathbf{N_R} \mid r(v, v') \in q\} = \{s_1, s_2, \ldots, s_n\}$. Similarly, we speak about concept/role names occurring in $q$. Note that the role conjunctions and concept/role names used in Definition 3.16 occur in $q$.

We will next link maximal fork rewritings and splittings. Let $q$ be a CQ and let $\mathsf{QTree}(\mathsf{maxfr}(q))$ denote the set of all forward-forest-shaped queries $\mathsf{maxfr}(q)\restriction_{\mathsf{Reach}(v)}$, where $v \in \mathrm{Var}(\mathsf{maxfr}(q))$ and $\mathsf{Reach}(v)$ denotes the set of all variables reachable from $v$ in $\mathcal{I}_{\mathsf{maxfr}(q)}$ via a directed path. Note that the size of $\mathsf{QTree}(\mathsf{maxfr}(q))$ is polynomial in the size of $\mathsf{maxfr}(q)$, thus also in $|q|$. The following lemma was shown in Appendix A by Lutz (2008b).

**Lemma 3.18.** *Let $\Pi^{\mathsf{N}}_{q'} = (\texttt{Roots}, \texttt{name}, \texttt{SubTree}_1, \ldots, \texttt{SubTree}_n, \texttt{root-of}, \texttt{Trees})$ be an $\mathsf{N}$-splitting of $q'$, a fork rewriting of a CQ $q$, let $q'_1, q'_2, \ldots, q'_k$ be the disconnected components of $q'\restriction_{\texttt{Trees}}$, and let $x_1, x_2, \ldots, x_n$ be the root variables of the corresponding $q'\restriction_{\texttt{SubTree}_i}$. Then:*

- $q'_i \in \mathsf{QTree}(\mathsf{maxfr}(q))$ *for all* $1 \leq i \leq k$,
- *for all* $1 \leq i \leq n$ *we have* $q'{\restriction_{\mathtt{SubTree}_i}} \in \mathsf{QTree}(\mathsf{maxfr}(q))$, *and*
- $\bigcap_{r \in \{r \mid r(\mathtt{root\text{-}of}(i), x_i) \in q'\}} .r$ *occurs in* $\mathsf{maxfr}(q)$.

*Proof.* A notational variant of a proof by Lutz (2008b, Lemma 4) assuming the naming convention from his Appendix A.[10] □

### 3.5 Step IV: Spoilers

Spoilers (Lutz, 2008b, p. 6) are $\mathcal{ALC}^{\cap}$ knowledge bases dedicated for "blocking" query matches over locally forward forest-like structures. We define them formally below.

**Definition 3.19.** *Let* $\mathsf{N} \subseteq \mathbf{N_I}$, *$q$ be a CQ and let*

$$\Pi_q^{\mathsf{N}} := (\mathtt{Roots}, \mathtt{name}, \mathtt{SubTree}_1, \ldots, \mathtt{SubTree}_n, \mathtt{root\text{-}of}, \mathtt{Trees})$$

*be an* $\mathsf{N}$*-splitting* $\Pi_q^{\mathsf{N}}$ *of* $q$. *An* $\mathsf{N}$-spoiler $\mathcal{K}^{\lightning}_{\Pi_q^{\mathsf{N}}}$ *for* $\Pi_q^{\mathsf{N}}$ *is an* $\mathcal{ALC}^{\cap}$*-KB satisfying at least one of:*

*(A)* $(\top \sqsubseteq \neg \mathrm{Match}_{\hat{q}}) \in \mathcal{K}^{\lightning}_{\Pi_q^{\mathsf{N}}}$ *for some forward-tree-shaped query* $\hat{q}$, *a connected component of* $q{\restriction_{\mathtt{Trees}}}$,

*(B)* $(\neg \mathrm{A}(\mathtt{name}(x))) \in \mathcal{K}^{\lightning}_{\Pi_q^{\mathsf{N}}}$ *for some atom* $\mathrm{A}(x) \in q$ *with* $x \in \mathtt{Roots}$,

*(C)* $(\neg r(\mathtt{name}(x), \mathtt{name}(y))) \in \mathcal{K}^{\lightning}_{\Pi_q^{\mathsf{N}}}$ *from some atom* $r(x, y) \in q$ *with* $x, y \in \mathtt{Roots}$,

*(D)* $\left(\neg \exists \left(\bigcap_{r(\mathtt{root\text{-}of}(i), x_i) \in q} r\right) .\mathrm{Match}_{q{\restriction_{\mathtt{SubTree}_i}}}\right) (\mathtt{name}(\mathtt{root\text{-}of}(i))) \in \mathcal{K}^{\lightning}_{\Pi_q^{\mathsf{N}}}$ *for some index* $1 \leq i \leq n$ *(where* $x_i$ *denotes the root variable of* $q{\restriction_{\mathtt{SubTree}_i}}$*).*

Observe a tight correspondence between Items (A)–(D) of the above definition and Items (A)–(D) from Definition 3.14. We may see these cases as potential ways of "blocking" the compatibility of a given splitting.

**Definition 3.20.** *Let* $\mathsf{N} \subseteq \mathbf{N_I}$ *and let* $q$ *be a CQ. An* $\mathcal{ALC}^{\cap}$*-KB* $\mathcal{K}^{\lightning\star}_q$ *is an* $\mathsf{N}$*-super-spoiler for* $q$ *if it is a* $\subseteq$*-minimal KB such that for all fork rewritings* $q'$ *of* $q$ *and all* $\mathsf{N}$*-splittings* $\Pi_{q'}^{\mathsf{N}}$ *of* $q'$ *we have that* $\mathcal{K}^{\lightning\star}_q$ *is an* $\mathsf{N}$*-spoiler for* $\Pi_{q'}^{\mathsf{N}}$.

The forthcoming lemmas show that the existence of an $\mathsf{N}$-super-spoiler "spoils" the (finite) entailment of an input CQ over (finite) $(|q|, \mathsf{N})$-lff-interpretations.

**Lemma 3.21.** *Let* $\mathcal{I}$ *be a (finite)* $(|q|, \mathsf{N})$*-lff-like interpretation and let* $q$ *be a CQ. Then* $\mathcal{I} \not\models q$ *implies that there is an* $\mathsf{N}$*-super-spoiler* $\mathcal{K}^{\lightning\star}_q$ *for* $q$ *such that* $\mathcal{I} \models \mathcal{K}^{\lightning\star}_q$.

*Proof.* We inductively construct a sequence $\mathcal{K}_0 := \emptyset, \mathcal{K}_1, \mathcal{K}_2, \ldots$ of $\mathcal{ALC}^{\cap}$-KBs converging to an $\mathsf{N}$-super-spoiler for $q$. To do so, we first fix some ordering on pairs $\mathrm{P}_i := (q', \Pi_{q'}^{\mathsf{N}})$ of fork rewritings $q'$ and $\mathsf{N}$-splittings of $q'$, and then proceed inductively with the $i$-th such pair. Note that $\Pi_{q'}^{\mathsf{N}}$ is not compatible with $\mathcal{I}$. Indeed, Lemma 3.17 would then imply $\mathcal{I} \models q$. Thus, there is at least one item of Definition 3.16 that is violated and there exists the

10. Watch out! There is a glitch in Lutz's proof. In the inductive assumption no. 4, there should be $\{v, v'\} \neq \{v, v'\} \cap S_j \neq \emptyset$ rather than $\{v, v'\} \cap S_j \neq \emptyset$.

corresponding axiom $\alpha$ for which $\mathcal{I} \not\models \alpha$ holds. Let $\beta$ be the corresponding "negated" axiom from Definition 3.19. We put $\mathcal{K}_i := \mathcal{K}_{i-1} \cup \{\beta\}$. From the definition of a spoiler and by construction, we see that $\mathcal{K}_i$ is an $\mathsf{N}$-spoiler for $\mathrm{P}_i$ as well as for all previous pairs. Moreover, $\mathcal{I} \models \beta$. There are only finitely many pairs $\mathrm{P}_i$ to consider, hence we can take $\mathcal{K}$ to be the last knowledge base on the list. Clearly $\mathcal{I} \models \mathcal{K}$. After replacing $\mathcal{K}$ with its $\subseteq$-minimal subset that entails $\mathcal{K}$, we obtain the desired $\mathsf{N}$-super-spoiler $\mathcal{K}_q^{\lightning\star}$ for $q$. □

**Lemma 3.22.** *Let $\mathcal{I}$ be a (finite) $(|q|, \mathsf{N})$-lfff-like interpretation and let $q$ be a CQ. Then if $\mathcal{I} \models \mathcal{K}_q^{\lightning\star}$ for some $\mathsf{N}$-super-spoiler $\mathcal{K}_q^{\lightning\star}$ for $q$ then $\mathcal{I} \not\models q$.*

*Proof.* Ad absurdum, suppose that $\mathcal{I} \models q$. Hence, by Lemma 3.17 we infer that there is a fork rewriting $q'$ of $q$ and an $\mathsf{N}$-splitting $\Pi_{q'}^{\mathsf{N}}$ of $q'$ that is compatible with $\mathcal{I}$. Since $\mathcal{K}_q^{\lightning\star}$ is an $\mathsf{N}$-super-spoiler for $q$ we have that, by Definition 3.20, it is also an $\mathsf{N}$-spoiler for $\Pi_{q'}^{\mathsf{N}}$. This implies that for $\Pi_{q'}^{\mathsf{N}}$ at least one of the conditions (A)–(D) from Definition 3.19 hold, contradicting the compatibility of $\Pi_{q'}^{\mathsf{N}}$ with $\mathcal{I}$. □

Relying on the presented lemmas we conclude a reduction from the (U)CQ entailment problem to the problem of checking the existence of an $\mathsf{ind}(\mathcal{K})$-super-spoiler spoiling the (finite) satisfiability of $\mathcal{K}$.

**Lemma 3.23.** *Let $\mathcal{DL}$ be (finitely) locally-forward abstract DL, $\mathcal{K}$ be a $\mathcal{DL}$-KB and $q := \bigvee_{i=1}^{m} q_i$ be a UCQ. Then $\mathcal{K} \not\models_{(\mathrm{fin})} q$ iff there are $\mathsf{ind}(\mathcal{K})$-super-spoilers $\mathcal{K}_{q_i}^{\lightning\star}$ for all $q_i$ s.t. $\mathcal{K} \cup \bigcup_i \mathcal{K}_{q_i}^{\lightning\star}$ is (finitely) satisfiable.*

*Proof sketch.* Similar to the proofs of Lemma 3.21 and Lemma 3.22. □

### 3.6 Step V: Super-Spoilers Made Small and Efficient

To get the optimal complexity bounds, we need to show that there are exponentially many super-spoilers that can be enumerated in exponential time and that the size of each super-spoiler is only of polynomial size.

We first show the following lemma (an analogue to Lutz's work (2008b, Lemma 5)), showing that super-spoilers have "small" sizes.

**Lemma 3.24.** *Let $\mathsf{N} \subseteq \mathbf{N_I}$ be finite, $q$ be a CQ, and let $\mathcal{K}_q^{\lightning\star}$ be an $\mathsf{N}$-super-spoiler for $q$. Then all the axioms contained in $\mathcal{K}_q^{\lightning\star}$ are of one of the following forms:*

*(A′)* $\top \sqsubseteq \neg\mathrm{Match}_{\hat{q}}$ *for some forward-tree-shaped query* $\hat{q} \in \mathsf{QTree}(\mathsf{maxfr}(q))$,

*(B′)* $\neg\mathrm{A}(\mathtt{a})$ *for some name* $\mathtt{a} \in \mathsf{N}$ *and a concept name* A *occurring in* $\mathsf{maxfr}(q)$,

*(C′)* $\neg r(\mathtt{a}, \mathtt{b})$ *for some names* $\mathtt{a}, \mathtt{b} \in \mathsf{N}$ *and a role name* $r$ *occurring in* $\mathsf{maxfr}(q)$,

*(D′)* $(\neg\exists\,(s_1 \cap s_2 \cap \ldots \cap s_k)\,.\mathrm{Match}_{\hat{q}})\,(\mathtt{a})$ *for some forward-tree-shaped* $\hat{q} \in \mathsf{QTree}(\mathsf{maxfr}(q))$, *name* $\mathtt{a} \in \mathsf{N}$ *and a role conjunction* $s_1 \cap s_2 \cap \ldots \cap s_k$ *that occurs in* $\mathsf{maxfr}(q)$.

*Proof.* Take any $\mathsf{N}$-super-spoiler for $q$ and let $\alpha$ be any of its axioms. We show that $\alpha$ is of the shape above. If $\alpha$ is of the form of Item (B) or Item (C) we are done by the fact that (1) if $r$/A occurs in $q$ then it also occurs in the maximal fork rewriting, and (2) `name` assigns values from $\mathsf{N}$. If $\alpha$ is of the form of Item (A) we invoke Lemma 3.18. Applying all mentioned arguments we are also done with the case when $\alpha$ has the form from Item (D). □

As a direct consequence of the above lemma we obtain:

**Lemma 3.25.** *The size of every* $\mathsf{N}$*-super-spoiler for a CQ* $q$ *is polynomial in* $|q| + |\mathsf{N}|$*, and the total number of* $\mathsf{N}$*-super-spoilers is exponential in* $|q| + |\mathsf{N}|$*.*

*Proof.* Let $q^* := \mathsf{maxfr}(q)$. To bound the size of $\mathsf{N}$-super-spoilers we invoke Lemma 3.24 and see that the axioms of the corresponding items can be bounded, respectively, by $|\mathsf{QTree}(q^*)|$, $|q| \cdot |\mathsf{N}|$, $|q| \cdot |\mathsf{N}|^2$ and $|q| \cdot |\mathsf{QTree}(q^*)| \cdot |\mathsf{N}|$. Since $|\mathsf{QTree}(q^*)|$ is bounded polynomially in $|q|$ we are done. The latter part is now immediate. □

The last property in our path leading to an algorithm solving the (U)CQ-entailment is the ability to enumerate $\mathsf{N}$-super-spoilers in exponential time.

**Lemma 3.26.** *The set of all* $\mathsf{N}$*-super-spoilers for a CQ* $q$ *can be enumerated in time exponential in* $|\mathsf{N}| + |q|$*.*

*Proof.* We enumerate $\mathsf{N}$-super-spoilers as follows. We first enumerate all $\mathcal{ALC}^{\cap}$-KBs containing only the axioms stated in Lemma 3.24 (requires time exponential in $|\mathsf{N}| + |q|$). To check if a knowledge-base is indeed an $\mathsf{N}$-super-spoiler, we go through all fork rewritings (there are exponentially many in $|q|$ of them) and all splittings for them (exponential in $|\mathsf{N}| + |q|$). Then we apply the definition of $\mathsf{N}$-spoilers to check if the considered knowledge-base indeed blocks the splitting, which can be performed, after fixing an $\mathsf{N}$-spoiler and an $\mathsf{N}$-splitting, in polynomial time. The execution times are multiplied, hence the overall algorithm works in time exponential in $|\mathsf{N}| + |q|$. □

### 3.7 Step VI: The Algorithm

We are now ready to present an algorithm for deciding (finite) (U)CQ entailment problem over (finitely) locally-forward DLs, that is worst-case optimal in many scenarios, *e.g.* in the case when the (finite) satisfiability problem for a given DL is ExpTime-complete. We present it in pseudocode below.

**Procedure 1:** (Finite) UCQ entailment for (finitely) locally-forward DLs

**Input:** UCQ $q := \bigvee_{i=1}^{m} q_i$ and $\mathcal{DL}$-KB $\mathcal{K}$, where $\mathcal{DL} \in \mathscr{C}_{\mathsf{lf}}$ ($\mathcal{DL} \in \mathscr{C}_{\mathsf{flf}}$ in the finite case).

1 If $\mathcal{K}$ is not (finitely) satisfiable **return** `True`. `// Checkable in` $\mathsf{SAT}_{\mathcal{DL}}(\mathsf{poly}(|\mathcal{K}|))$.

2 **foreach** *selection of* $\mathsf{ind}(\mathcal{K})$*-super spoilers* $\mathcal{K}^{\blacktriangledown}_{q_1}, \ldots, \mathcal{K}^{\blacktriangledown}_{q_m}$ *for* $q_1, \ldots, q_m$ `// In` $\exp(|\mathsf{ind}(\mathcal{K})|+|q|)$ `by Lemma 3.26`

3 **do**

4 If $\mathcal{K} \cup \bigcup_i \mathcal{K}^{\blacktriangledown}_{q_i}$ is (finitely) satisfiable **return** `False`. `// In` $\mathsf{SAT}_{\mathcal{DL}}(\mathsf{poly}(|\mathcal{K}| + |q|))$ `by Lemma 3.25`

5 **return** `True`.

**Lemma 3.27.** *Procedure 1 returns* `True` *iff* $\mathcal{K} \models_{(\mathrm{fin})} q$*. Moreover, Procedure 1 can be implemented to work in time* $\mathsf{exp}(|\mathcal{K}| + |q|) \cdot \mathsf{SAT}_{\mathcal{DL}}(\mathsf{poly}(|\mathcal{K}|+|q|))$ *for some polynomial function* poly *and an exponential function* exp*, where* $\mathsf{SAT}_{\mathcal{DL}}$ *denotes the worst-case optimal running time of the (finite) satisfiability problem for* $\mathcal{DL}$*-KBs.*

*Proof sketch.* The first statement follows immediately from the pseudocode and Lemma 3.23. The second part of the lemma follows from Lemma 3.25 and Lemma 3.26. □

Relying on the above lemma, we conclude our main theorem.

**Theorem 3.28.** *For any (finitely) locally-forward DL $\mathcal{DL}$ with (finite) $\mathcal{DL}$-KB-satisfiability problem decidable in time $\mathsf{SAT}_{\mathcal{DL}}(\cdot)$, there exists a polynomial and an exponential function* poly *and* exp *such that the (finite) UCQ-entailment problem over $\mathcal{DL}$-KB is decidable, for an input $\mathcal{K}, q$, in time $\mathsf{exp}(|\mathcal{K}| + |q|) \cdot \mathsf{SAT}_{\mathcal{DL}}(\mathsf{poly}(|\mathcal{K}|+|q|))$.*

The most important application of our work is when the (finite) knowledge base satisfiability problem for $\mathcal{DL}$ is ExpTime-complete. The function $\mathsf{exp}(|\mathcal{K}|+|q|)\cdot\mathsf{SAT}_{\mathcal{DL}}(\mathsf{poly}(|\mathcal{K}|+|q|))$ reduces then to a single exponential function, and hence, we have the following corollary (where the corresponding lower bound follows from $\mathcal{ALC}$ (Schild, 1991, Prop. 3)).

**Corollary 3.29.** *The (finite) (U)CQ entailment problem is ExpTime-complete for any (finitely) locally-forward abstract DL with ExpTime-complete (finite) knowledge base satisfiability problem.*

Recall from the beginning of the section that $\mathcal{ALCSCC}$+ERCBoxes is finitely locally-forward (to be shown in Corollary 4.24) and that any abstract DL $\mathcal{DL}$ contained in $\mu\mathcal{ALCSCC}$ is locally-forward, *cf.* Corollary 3.6. Their corresponding knowledge base satisfiability problems are ExpTime-complete (*cf.* Theorem 7 by Baader et al. (2020) and Corollary 3.6).

**Corollary 3.30.** *The UCQ entailment problem over $\mathcal{DL}$-KBs for any $\mathcal{DL}$ contained in $\mu\mathcal{ALCSCC}$ is ExpTime-complete. The finite UCQ entailment problem for $\mathcal{ALCSCC}$ w.r.t ERCBoxes is ExpTime-complete.*

Corollary 3.30 closes numerous gaps in the complexity of query entailment that were present in the literature, *e.g.* the complexities of CQ entailment for $\mathcal{ALC}b$, $\mathcal{ALCH}b_{\mathsf{reg}}\mathcal{Q}$, $\mu\mathcal{ALC}$, or the UCQ entailment for $\mathcal{ALCHQ}$ were, up to our knowledge, unknown. This also proves that neither regular role expressions nor safe boolean role combinations from $\mathcal{ALCH}b_{\mathsf{reg}}\mathcal{Q}$ do increase the complexity of querying, as it is the case for nominals and inverses. Call $\mathcal{Z}^-$ the logic $\mathcal{ALCH}b_{\mathsf{reg}}$ (this is the logic $\mathcal{Z}$ (Calvanese et al., 2009) without the Self operator). It was recently shown that $\mathcal{ZOQ}$ is finitely controllable (Bednarczyk & Kieroński, 2022, Theorem 3.1), hence by applying Corollary 3.30 we infer:

**Corollary 3.31.** *The finite UCQ entailment problem over $\mathcal{DL}$-KBs for any $\mathcal{ALC} \subseteq \mathcal{DL} \subseteq \mathcal{Z}^-\mathcal{Q}$ is ExpTime-complete.*

As the last remark: any PEQ can be transformed into a (U)CQ of exponential size. This yields us 2ExpTime-completeness of PEQ querying for any logics mentioned in the above theorem. The lower bound holds already for $\mathcal{ALC}$ (Ortiz & Simkus, 2014, Theorem 1).

## 4. Sufficient Conditions for an Abstract DL to be "Locally-Forward"

This section is dedicated to providing model-theoretic constructions and sufficient conditions useful for determining whether a given abstract DL $\mathcal{DL}$ is locally-forward. Our aim is to make such conditions easier to check than applying Definition 3.4 directly.

Our agenda is as follows. We first introduce a concept of *forward unravellings* that turn interpretations into (usually infinite) forward forests. If the satisfaction of $\mathcal{DL}$-KBs is preserved under such unravellings, then $\mathcal{DL}$ is locally-forward. To obtain suitable conditions in the finite model reasoning scenario, we develop the notion of *scattered forward unravellings*. Their main advantage is that the scattered forward unravellings of finite interpretations are indeed finite whilst from the queries' view point they are indistinguishable from infinite forward forests. We conclude that $\mathcal{DL}$ is finitely locally-forward whenever the satisfaction of any $\mathcal{DL}$-KB is preserved under scattered forward unravellings. Finally, we will discuss the case of logics expressing statistical constraints over the domain, and see how the notion of scattered forward unravellings can be adjusted to them.

### 4.1 Forward unravellings of interpretations

Forward unravellings make interpretations forward-forest-shaped. Such a notion differs only slightly from classical unravellings (Baader et al., 2017, Definition 3.21). The crucial difference is that forward unravellings preserve substructures induced by the selected named individuals. More precisely, the sequences starting with two named individuals are excluded from the domain and roles linking named individuals are assigned "manually"(*cf.* the last item from Definition 4.1). Before reading the definition we suggest consulting Figure 3.

**Definition 4.1.** *Let* $\mathsf{N} \subseteq \mathbf{N_I}$ *be non-empty. An* $\mathsf{N}$-rooted forward unravelling *of an interpretation* $\mathcal{I}$ *is an interpretation* $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}} := (\Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}, \cdot^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}})$ *satisfying all four conditions below:*

1. *The domain of* $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ *consists of all non-empty words of elements from* $\Delta^{\mathcal{I}}$*, except those, where the first two elements are named or where two consecutive elements are "disconnected". In symbols:* $\Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} := (\Delta^{\mathcal{I}})^+ \setminus$

$$\Big( \mathsf{N}^{\mathcal{I}} \cdot \mathsf{N}^{\mathcal{I}} \cdot (\Delta^{\mathcal{I}})^* \ \cup \ (\Delta^{\mathcal{I}})^* \cdot \{\mathrm{de} \mid \mathrm{d}, \mathrm{e} \in \Delta^{\mathcal{I}}, \mathsf{RoI}_{\mathcal{I}}(\mathrm{d}, \mathrm{e}) = \emptyset\} \cdot (\Delta^{\mathcal{I}})^* \Big).$$

   *For convenience, we do not syntactically distinguish elements from* $\Delta^{\mathcal{I}}$ *and single-letter words from* $\Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$*. In particular, this means that* $\Delta^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$*. We often use* $\mathsf{last}$*,* i.e. *the function that maps a sequence to its last element.*

2. *For all individual names* $\mathtt{a} \in \mathsf{N}$ *we have* $\mathtt{a}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} = \mathtt{a}^{\mathcal{I}}$ *and for all other names* $\mathtt{a} \in (\mathbf{N_I} \setminus \mathsf{N})$ *there is some* $\mathtt{b} \in \mathsf{N}$ *fulfilling* $\mathtt{a}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} = \mathtt{b}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$*.*

3. *For any concept name* $\mathrm{A} \in \mathbf{N_C}$ *the equality* $\mathrm{A}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} = \{w \mid \mathsf{last}(w) \in \mathrm{A}^{\mathcal{I}}\}$ *holds.*

4. *For all role names* $r \in \mathbf{N_R}$ *we define* $r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ *as the intersection of* $\Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} \times \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ *and the set*

$$\Big(r^{\mathcal{I}} \cap (\mathsf{N}^{\mathcal{I}} \times \mathsf{N}^{\mathcal{I}})\Big) \cup \Big\{(w, w \cdot \mathrm{d}) \mid (\mathsf{last}(w), \mathrm{d}) \in r^{\mathcal{I}}\Big\}.$$

   *Intuitively, the above set is composed of two parts, where the first component preserves* $\mathsf{N}$*-named parts, while the other one preserves roles, mimicking the classical unravelling.*

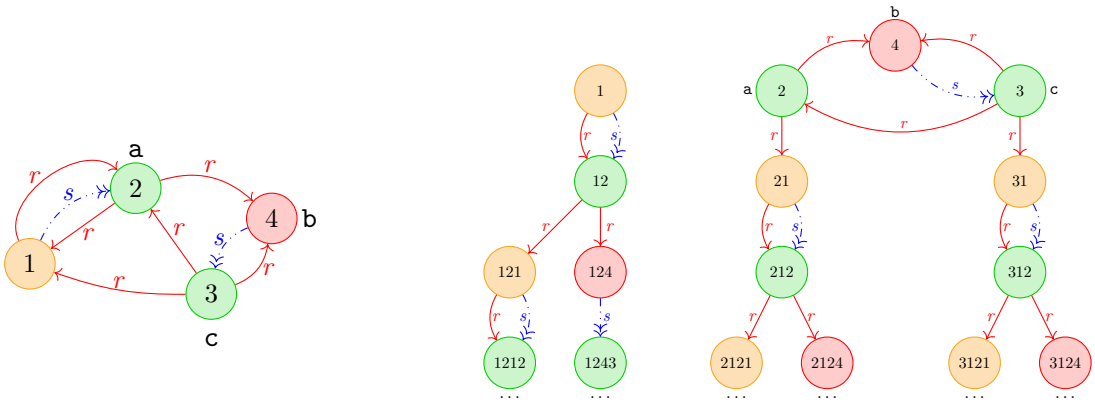



Figure 3: An interpretation $\mathcal{I}$ (left) with a fragment of its $\{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$-*rooted forward unravelling* $\mathcal{I}^{\vec{\omega}}_{\{\mathsf{a},\mathsf{b},\mathsf{c}\}}$ (right). Note that $\mathcal{I}^{\vec{\omega}}_{\{\mathsf{a},\mathsf{b},\mathsf{c}\}}$ is a forest with two connected components. Nodes of the same colour satisfy the same atomic concepts.

It is not difficult to see that forward unravellings produce forest-shaped interpretations that can be homomorphically mapped to the original structures.

**Lemma 4.2.** *Let $\mathcal{I}$ be an interpretation, $\mathsf{N} \subseteq \mathbf{N_I}$ be a set of names and let $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$ be any $\mathsf{N}$-rooted forward unravelling of $\mathcal{I}$. Then $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$ is $\mathsf{N}$-rooted forest-shaped and the function $\mathsf{last} : \Delta^{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}} \to \Delta^{\mathcal{I}}$ is an $\mathsf{N}$-homomorphism from $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$ to $\mathcal{I}$.*

*Proof sketch.* Routine case analysis, essentially unfolding Definition 4.1. □

This leads us to a sufficient condition for a DL to be locally-forward. Call an abstract DL $\mathcal{DL}$ *preserved under forward unravellings* if for all $\mathcal{DL}$-KBs $\mathcal{K}$ the $\mathsf{ind}(\mathcal{K})$-rooted forward unravellings are consistency-preserving, *i.e.* for all $\mathcal{I}$ we have $\mathcal{I} \models \mathcal{K}$ implies $\mathcal{I}^{\vec{\omega}}_{\mathsf{ind}(\mathcal{K})} \models \mathcal{K}$.

**Theorem 4.3.** *If an abstract DL $\mathcal{DL}$ is preserved under forward unravellings then $\mathcal{DL} \in \mathscr{C}_{\mathsf{lf}}$.*

*Proof.* By unfolding Definition 3.4 and Definition 3.3, it suffices to take any satisfiable $\mathcal{DL}$-KB $\mathcal{K}$, any of its models $\mathcal{I}$ and any $n \in \mathbb{N}$ and show that there exists an $(n, \mathsf{ind}(\mathcal{K}))$-lff-like model $\mathcal{J} \models \mathcal{K}$, whose all $n$-neighbourhoods can be $\mathsf{ind}(\mathcal{K})$-homomorphically-mapped to $\mathcal{I}$. Take $\mathcal{J} := \mathcal{I}^{\vec{\omega}}_{\mathsf{ind}(\mathcal{K})}$, and take any of its $n$-neighbourhoods $\mathcal{J}'$. Since $\mathsf{last}$ is an $\mathsf{ind}(\mathcal{K})$-homomorphism from $\mathcal{J}$ to $\mathcal{I}$, it is also an $\mathsf{ind}(\mathcal{K})$-homomorphism from $\mathcal{J}'$ to $\mathcal{I}$. □

We conclude by presenting several useful properties of forward unravellings that may be useful for a quick test to see whether the modelhood of some knowledge base is preserved.

**Property 4.4.** *Let $\mathcal{I}$ be an interpretation, let $\mathsf{N} \subseteq \mathbf{N_I}$ be a set of names and let $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$ be any $\mathsf{N}$-rooted forward unravelling of $\mathcal{I}$. Then the following conditions are satisfied:*

*(A) The interpretations $\mathcal{I}$ and $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$ restricted to all $\mathsf{N}$-named elements are isomorphic.*

*(B) For any concept name $\mathrm{A} \in \mathbf{N_C}$ we have that $\mathrm{A}^{\mathcal{I}}$ is non-empty iff $\mathrm{A}^{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}$ is. Similarly, for any role name $r \in \mathbf{N_R}$ we have that $r^{\mathcal{I}}$ is non-empty iff $r^{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}$ is.*

*(C) For any* $w \in \Delta^{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}$ *we have* $\mathsf{Conc}_{\mathcal{I}}(\mathsf{last}(w)) = \mathsf{Conc}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(w)$*. Moreover, for all* $v$ *satisfying* $(w, v) \in r^{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}$ *for some* $r \in \mathbf{N_R}$*, we have* $\mathsf{Rol}_{\mathcal{I}}(\mathsf{last}(w), \mathsf{last}(v)) = \mathsf{Rol}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(w, v)$*.*

*(D) For any* $w \in \Delta^{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}$ *we have that* $w$ *in* $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$ *and* $\mathsf{last}(w)$ *in* $\mathcal{I}$ *have the same number of successors satisfying the same roles and concepts. Formally, for any* non-empty *set of role names* $\mathsf{R} \subseteq \mathbf{N_R}$ *and any set of concept names* $\mathsf{C} \subseteq \mathbf{N_C}$ *we have that the cardinalities of the two sets below coincide:*

$$\Big\{v \in \Delta^{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}} \mid \mathsf{Conc}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(v) = \mathsf{C} \text{ and } \mathsf{Rol}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(w, v) = \mathsf{R}\Big\},$$

$$\Big\{\mathrm{e} \in \Delta^{\mathcal{I}} \mid \mathsf{Conc}_{\mathcal{I}}(\mathrm{e}) = \mathsf{C} \text{ and } \mathsf{Rol}_{\mathcal{I}}(\mathsf{last}(w), \mathrm{e}) = \mathsf{R}\Big\}.$$

*(E) For any* $w \in \Delta^{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}$ *we have that* $w$ *and* $\mathsf{last}(w)$ *are* directed-path-equivalent*, that is:*

- *If* $\rho$ *with* $\rho_1 := w$ *is a (possibly infinite) directed path in* $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$ *then* $\rho'$*, defined as* $\rho'_i := \mathsf{last}(\rho_i)$ *for all* $i$*, is a directed path in* $\mathcal{I}$ *such that for all* $i$ *we have* $\mathsf{Conc}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(\rho_i) = \mathsf{Conc}_{\mathcal{I}}(\rho'_i)$*, and for all* $i > 1$ *we have* $\mathsf{Rol}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(\rho_{i-1}, \rho_i) = \mathsf{Rol}_{\mathcal{I}}(\rho'_{i-1}, \rho'_i)$*.*
- *If* $\rho$ *with* $\rho_1 := \mathsf{last}(w)$ *is a (possibly infinite) directed path in* $\mathcal{I}$ *then* $\rho'$ *defined as:*
  *(i)* $\rho'_1 := w$*, and*
  *(ii)* $\rho'_i := \rho_i$ *if both* $\rho'_{i-1}$ *and* $\rho_i$ *are* $\mathsf{N}$*-named, and* $\rho'_i := \rho'_{i-1}\rho_i$ *otherwise for all remaining* $i$*,*

  *is a directed path in* $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$ *such that for all* $i$ *we have* $\mathsf{Conc}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(\rho'_i) = \mathsf{Conc}_{\mathcal{I}}(\rho_i)$*, and for all* $i > 1$ *we have* $\mathsf{Rol}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(\rho'_{i-1}, \rho'_i) = \mathsf{Rol}_{\mathcal{I}}(\rho_{i-1}, \rho_i)$*.*

*Proof sketch.* Follows immediately from the construction of $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$, *cf.* Definition 4.1. □

One can employ Property 4.4 to show that forward unravellings, among other properties, preserve: the satisfaction of ABoxes (via Item (A)), the existence of "unary-types" and "role-types" (via Item (B)), cardinality constraints on the total number of successors (via Item (D)), role hierarchies and safe boolean role combinations (via Item (C)), regular role expressions and fixed points (via Item (E)). Hence, they can be easily used to give a self-contained proof of Corollary 3.6 saying that the logic $\mu\mathcal{ALCSCC}$ is locally forward. We leave as an exercise to the reader to see that (in most of the cases) our unravellings do not preserve inverse roles, nominals, transitivity, or self-loops.

### 4.2 Scattered Forward Unravellings of Interpretations

We have seen that for a certain class of logics, namely for DLs preserved under forward unravellings, forward unravellings produce forest (counter)models out of (possibly) non-forest ones. The presented construction is sufficient to employ Lutz's spoiler method (described in the previous section) over the class of *arbitrary* structures, but it is useless once we want to achieve results in the finite-model scenario. The reason is trivial: forward unravellings of finite interpretations are nearly always infinite. To find a suitable counterpart of forward unravellings in the finite realm, we design the notion of *scattered forward unravellings*.

The aim of scattered forward unravellings is, given a threshold $n$, a set of names $\mathsf{N}$, and a finite interpretation $\mathcal{I}$, to turn $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ into a finite $(n,\mathsf{N})$-lff-like interpretation (consult again Definition 3.1 if needed). The construction is a little bit involved and relies on cutting out finitely many tree-like components from $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ and then glueing them together mimicking "parent-to-leaf connections", so that any neighbourhood of size $n$ from the desired structure is homomorphically-equivalent to some $n$-neighbourhood from the forward unravelling. The novel model construction presented here took some inspiration from similar constructions, namely the ones by Bednarczyk and Kieroński (2022, Sec. 3), by Otto (2004, Sec. 4.2), and by Emerson and Halpern (1985, Sec. 3.5).

We start the construction by defining basic building blocks, called here the *components*.

**Definition 4.5.** *Fix a number $n \in \mathbb{N}$, a finite set of names $\mathsf{N} \subseteq \mathbf{N_I}$ and a finite interpretation $\mathcal{I}$ whose domain is linearly ordered. We select certain auxiliary substructures of $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$.*

- *The* $(\mathsf{N},n)$-king component $\mathcal{I}_{\underline{\text{♚}}}$ *of $\mathcal{I}$ is obtained from $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ by a restriction to all words from $\Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ of length at most $2n{+}1$.*
- *Call* $\mathrm{d} \in \Delta^{\mathcal{I}}$ deep *whenever there is a word $w := u \cdot \mathrm{d}$ in $\Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ such that $|u| > n$. The lexicographically smallest word such $w$ is called the* deep realisation of $\mathrm{d}$. *The* $(\mathsf{N},n)$-pawn component $\mathcal{I}_{\underline{\text{♟}},\mathrm{d}}$ *of a deep element* $\mathrm{d}$ *is obtained by restricting $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ to all words of the from $w \cdot v$, where $w$ is the deep realisation of* $\mathrm{d}$ *and $|v| \leq 2n$.*

*Let $\mathscr{I}(n,\mathsf{N},\mathcal{I})$ be the set of all components of $\mathcal{I}$. With $L(n,\mathsf{N},\mathcal{I})$ we denote the maximal number of leaves among all components in $\mathscr{I}(n,\mathsf{N},\mathcal{I})$. Thus we can assign a number $1 \leq \ell \leq L(n,\mathsf{N},\mathcal{I})$ to any leaf in any component of $\mathscr{I}(n,\mathsf{N},\mathcal{I})$ and refer to it as this component's $\ell$-th leaf. For future purposes, we define $\mathsf{orig}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} : (\bigcup \mathscr{I}(n,\mathsf{N},\mathcal{I})) \to \mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ that maps an element from any component to the element from whom it originated. We also employ $\mathsf{orig}_{\mathcal{I}} := \mathsf{orig}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} \circ \mathsf{last}$.*

The king component $\mathcal{I}_{\underline{\text{♚}}}$ is simply the forward unravelling cut off after $2n{+}1$ steps, while each pawn component $\mathcal{I}_{\underline{\text{♟}},\mathrm{d}}$ is a subtree of depth $2n$ rooted at some deep enough copy of some element of $\Delta^{\mathcal{I}}$. Note that $\mathcal{I}_{\underline{\text{♚}}}$ is a finite $\mathsf{N}$-rooted forward forest, while all $\mathcal{I}_{\underline{\text{♟}},\mathrm{d}}$ are forward trees. As the next step, we provide sufficiently many copies of interpretations from $\mathscr{I}(n,\mathsf{N},\mathcal{I})$.

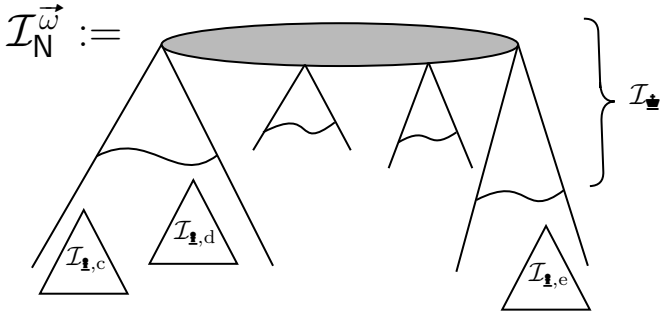



Figure 4: Visualization of the selection of components out of the forward unravelling of $\mathcal{I}$.

**Definition 4.6.** *Take $(n,\mathsf{N},\mathcal{I})$ as in Definition 4.5. The set $\mathscr{J}(n,\mathsf{N},\mathcal{I})$ of* copies of components *is composed of the king component $\mathcal{I}_{\underline{\text{♚}}} \in \mathscr{I}(n,\mathsf{N},\mathcal{I})$ and isomorphic copies $\mathcal{I}_{\underline{\text{♟}},\mathrm{d},h}^{(\ell,\mathrm{s})}$ of pawn components $\mathcal{I}_{\underline{\text{♟}},\mathrm{d}}$ that are indexed by: a deep element $\mathrm{d} \in \Delta^{\mathcal{I}}$, $h \in \{0,1\}$, $1 \leq \ell \leq$*

$L(n, \mathsf{N}, \mathcal{I})$, *and* $\mathrm{s} \in \left(\Delta^{\mathcal{I}} \cup \{\blacksquare\}\right)$, *that are called, respectively, the* target, *the* hue, *the* leaf number, *and the* source *of a component. We often employ* $*$ *that serves as a wildcard which can be used in place of various parameters, whenever such parameters are of no importance. For convenience, we will also speak about the "hue" of a domain element, meaning the "hue" of the unique copy of the pawn component to which such an element belongs.*

The main idea behind the quite loaded notation of $\mathcal{I}^{(\ell,\mathrm{s})}_{\blacksquare,\mathrm{d},h}$ is that the root of $\mathcal{I}^{(\ell,\mathrm{s})}_{\blacksquare,\mathrm{d},h}$ "can serve as a d-witness for the $\ell$-th leaf of any copy of $\mathcal{I}_{\blacksquare,\mathrm{s}}$ of hue $1-h$". Note that it follows immediately from the finiteness of $\Delta^{\mathcal{I}}$ and $n$, that $\mathscr{J}(n, \mathsf{N}, \mathcal{I})$ is finite.

We are finally ready to present the definition of $n$-scattered forward unravellings.

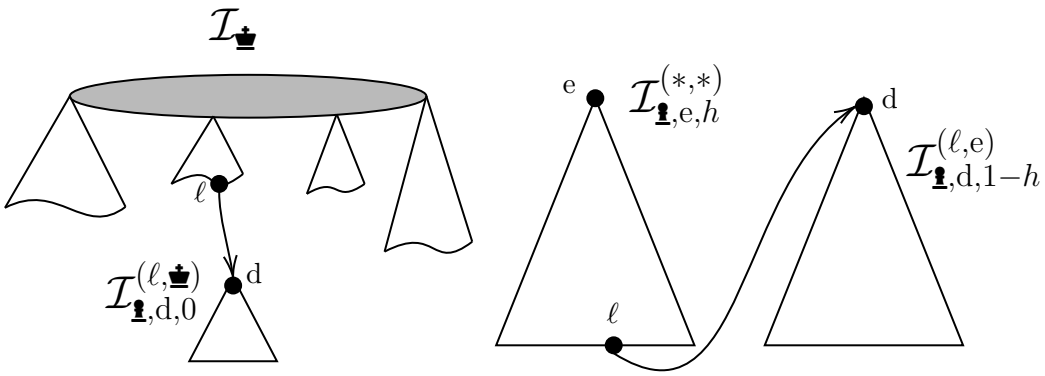



Figure 5: Linking components in the construction of $(n, \mathsf{N})$-scattered unravelling of $\mathcal{I}$. The left hand side of the picture illustrates connections between the king component and pawn components, while its right hand side depicts connections between two pawn components.

**Definition 4.7.** *Suppose that* $(n, \mathsf{N}, \mathcal{I})$ *are as in Definition 4.5. The* $(n, \mathsf{N})$-scattered-forward-unravelling $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ *is obtained by taking the disjoint sum of structures from* $\mathscr{J}(n, \mathsf{N}, \mathcal{I})$, *and then extending the interpretation of each role name* $r \in \mathbf{N_R}$, *with a pair* $(u, v)$ *of* $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$, *whenever:* $(\mathsf{orig}_{\mathcal{I}}(u), \mathsf{orig}_{\mathcal{I}}(v)) \in r^{\mathcal{I}}$ *holds and either*

*(i)* $u$ *is the* $\ell$*-th leaf of* $\mathcal{I}_{\blacksquare}$, *and* $v$ *is the root of* $\mathcal{I}^{(\ell,\blacksquare)}_{\blacksquare,\mathsf{orig}_{\mathcal{I}}(v),0}$, *or*

*(ii)* $u$ *is the* $\ell$*-th leaf of* $\mathcal{I}^{(*,*)}_{\blacksquare,\mathrm{e},h}$ *for an* $\mathrm{e} \in \Delta^{\mathcal{I}}$ *and* $h \in \{0, 1\}$, *and* $v$ *is the root of* $\mathcal{I}^{(\ell,\mathrm{e})}_{\blacksquare,\mathsf{orig}_{\mathcal{I}}(v),1-h}$.

Our next goal is to show that the aforementioned construction fulfils its purposes. *i.e.* that $(\mathsf{N}, n)$-scattered forward unravellings of *finite* structures are indeed finite (which we already discussed) and that they are $(n, \mathsf{N})$-locally-forest-like. As the first step we establish:

**Lemma 4.8.** *Assume* $(n, \mathsf{N}, \mathcal{I})$ *are as in Definition 4.5. Then every undirected path in* $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ *leading from a root of some pawn component to any of its leaves has length* $\geq 2n$.

*Proof sketch.* By construction of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$, the path $\rho$ crosses more than one component and does not contain two consecutive positions being roots of different components. Then $\rho$ has the shape $\rho'_0 \mathrm{d}_1 \dots \rho'_{2k} \mathrm{d}_{2k+1}$, where the even-numbered paths $\rho'_{2i}$ are (possibly single-element) leaf-to-leaf paths traversing a single component, while the odd-numbered elements $\mathrm{d}_{2i+1}$ are roots of components. Moreover, if $\rho_i$ is a leaf and $\rho_{i+1}$ is a root, then their "hues" are different. This yields a contradiction, since the first and the last element of $\rho$ are of different "hue", and thus clearly cannot be in the same component. □

Call a neighbourhood *safe* if it is fully contained in some (copy of a) component. The above lemma, together with the definition of the king component, yield a general overview on how $n$-neighbourhoods interact with components in scattered unravellings.

**Corollary 4.9.** *Let $n$, $\mathsf{N}$, $\mathcal{I}$ be as in Definition 4.5, and let $\mathcal{N}$ be any unsafe $n$-neighbourhood of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$. Then $\mathcal{N}$ does not contain any $\mathsf{N}$-named elements. Moreover, for all components $\mathcal{J}$ of $\mathscr{J}(n, \mathsf{N}, \mathcal{I})$ sharing a common element with $\mathcal{N}$, we have the dichotomy: either $\mathcal{N}$ contains some leaf of $\mathcal{J}$ (we call such component $\mathcal{N}$-*upper*) or $\mathcal{N}$ contains the root of $\mathcal{J}$ (we call such component $\mathcal{N}$-*lower*).*

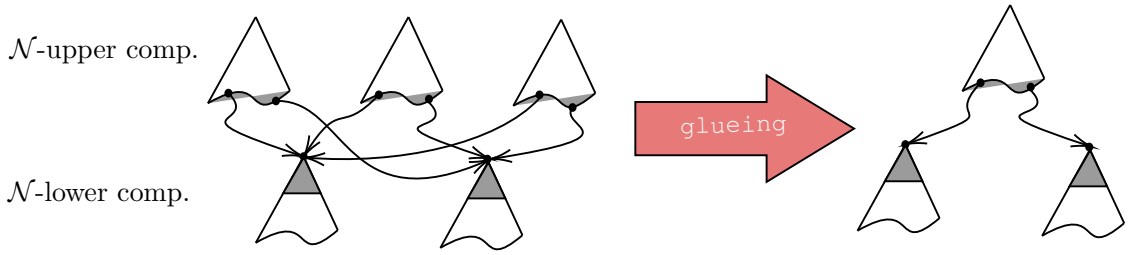



Figure 6: Visualisation of $\mathcal{N}$-lower and $\mathcal{N}$-upper components of some $n$-neighbourhood $\mathcal{N}$ of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$. The right part of the picture presents the notion of glueing, to be defined later.

As the second step, we show that all $\mathcal{N}$-upper components look alike.

**Lemma 4.10.** *Let $(n, \mathsf{N}, \mathcal{I})$ be as in Definition 4.5, and take any unsafe $n$-neighbourhood $\mathcal{N}$ of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$. Then either the only $\mathcal{N}$-upper component is the king component, or there exists an element $\mathrm{d} \in \Delta^{\mathcal{I}}$ such that every $\mathcal{N}$-upper component is of the from $\mathcal{I}^{(*,*)}_{\mathbb{1},\mathrm{d},*}$. In particular, this means that all $\mathcal{N}$-upper components are isomorphic.*

*Proof sketch.* Follows from the proof scheme of Lemma 4.8, by analysing the shape of $\rho$. □

Given an unsafe $n$-neighbourhood $\mathcal{N}$ of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$, a *glueing* $\mathsf{glue}(\mathcal{I})$ of $\mathcal{N}$ is defined as the restriction of $\mathcal{N}$ to all elements from all $\mathcal{N}$-lower components and all elements from one *single* $\mathcal{N}$-upper component. By Lemma 4.10 all $\mathcal{N}$-upper components are isomorphic copies of the same component, thus *the* glueing of $\mathcal{N}$ is unique up to isomorphism. It follows from the construction of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ that the glueing of $\mathcal{N}$ is a forward tree. Moreover, the identity function is clearly a homomorphism from $\mathsf{glue}(\mathcal{N})$ to $\mathcal{N}$. A homomorphism in the other direction is established next.

**Lemma 4.11.** *Let $(n, \mathsf{N}, \mathcal{I})$ be as in Definition 4.5, and take any unsafe $n$-neighbourhood $\mathcal{N}$ of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$. Then the glueing $\mathsf{glue}(\mathcal{N})$ of $\mathcal{N}$ is a forward tree, homomorphically equivalent to $\mathcal{N}$.*

*Proof sketch.* Show that a function $\mathfrak{h} : \mathcal{N} \to \mathsf{glue}(\mathcal{N})$, defined as the identity on $\mathcal{N}$-lower components, and as a mapping of elements from $\mathcal{N}$-upper components to their corresponding elements in the unique $\mathsf{glue}(\mathcal{N})$-upper component is the desired homomorphism. This follows by a case analysis. The key property is that whenever the $\ell$-th leaf $\mathrm{d}'$ of some $\mathcal{I}^{(*,*)}_{\mathbb{1},\mathrm{c},h}$ component satisfies $(\mathrm{d}', \mathrm{e}) \in r^{\mathcal{N}}$ for a root of some $\mathcal{N}$-lower component then *all* $\ell$-th leaf $\mathrm{d}'$ from all $\mathcal{I}^{(*,*)}_{\mathbb{1},\mathrm{c},h}$ component satisfy $(\mathrm{d}', \mathrm{e}) \in r^{\mathcal{N}}$. The other parts of the proof were already discussed. □

As an immediate corollary of the previous lemma we conclude the following.

**Corollary 4.12.** *For any finite $n \in \mathbb{N}$, any set of names $\mathsf{N} \subseteq \mathbf{N_I}$, and any finite interpretation $\mathcal{I}$, every $(n, \mathsf{N})$-scattered-forward-unravelling $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ of $\mathcal{I}$ is $(n, \mathsf{N})$-lff-like.*

We say that an abstract DL $\mathcal{DL}$ is *preserved under scattered forward unravellings* if for all finitely satisfiable $\mathcal{DL}$-KBs $\mathcal{K}$, their finite models $\mathcal{I}$, and for all but finitely-many positive integers $n \in \mathbb{N}$, the $n, \mathsf{ind}(\mathcal{K})$-scattered forward unravelling $\mathcal{I}^{\vec{n}}_{\mathsf{ind}(\mathcal{K})}$ of $\mathcal{I}$ is a finite model of $\mathcal{K}$. Relying on Corollary 4.12 one can finally show that any DL preserved under scattered forward unravellings is also finitely locally forward. Indeed:

**Theorem 4.13.** *If a description logic $\mathcal{DL}$ is preserved under scattered forward unravellings then $\mathcal{DL}$ belongs to $\mathscr{C}_{\mathsf{flf}}$.*

*Proof sketch.* The proof is analogous to the proof of Theorem 4.3. □

Similarly to the end of the previous section, we conclude with a list of useful properties of scattered forward unravellings. The idea is that for future applications one can employ Property 4.14 in order to show preservation of: satisfaction of ABoxes (via Item (A)), existence of "unary-types" and "role-types" (via Item (B)), cardinality constraints on the total number of successors (via Item (D)), role hierarchies and safe boolean role combinations (via Item (C)), regular role expressions and fixed points (via Item (E)). Consult the following lemma:

**Property 4.14.** *Assume the parameters $(n, \mathsf{N}, \mathcal{I})$ as in Definition 4.5 and let $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ be any $(n, \mathsf{N})$-scattered-forward-unravelling of $\mathcal{I}$. Then the following conditions are satisfied:*

*(A) The interpretations $\mathcal{I}$ and $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ restricted to all $\mathsf{N}$-named elements are isomorphic.*

*(B) For any concept name $\mathrm{C}$ we have that $\mathrm{C}^{\mathcal{I}}$ is non-empty iff $\mathrm{C}^{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}$ is non-empty. Similarly, for any role name $r$ we have that $r^{\mathcal{I}}$ is non-empty iff $r^{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}$ is non-empty.*

*(C) For any $w \in \Delta^{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}$ we have $\mathsf{Conc}_{\mathcal{I}}(\mathsf{orig}_{\mathcal{I}}(w)) = \mathsf{Conc}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(w)$. Moreover, for all $v$ satisfying $(w, v) \in r^{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}$ for some $r \in \mathbf{N_R}$, we have $\mathsf{Rol}_{\mathcal{I}}(\mathsf{orig}_{\mathcal{I}}(w), \mathsf{orig}_{\mathcal{I}}(v)) = \mathsf{Rol}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(w, v)$.*

*(D) For all $w \in \Delta^{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}$ we have that $w$ in $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ and $\mathsf{orig}_{\mathcal{I}}(w)$ in $\mathcal{I}$ have the same number of successors satisfying the same roles and concepts,* i.e. *an analogue of Item (D) of Property 4.4 holds.*

*(E) For any $w \in \Delta^{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}$ we have that $w$ and $\mathsf{orig}_{\mathcal{I}}(w)$ are directed-path-equivalent,* i.e. *an analogue of Item (E) of Property 4.4 holds.*

*Proof sketch.* Follows by case-analysing the construction of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ and employing Property 4.4. □

### 4.3 Scattered Unravellings with Rebalancing

Some description logics can express both local cardinality constraints (*i.e.* constraints concerning the role successors of specific individuals) and global cardinality constraints (*i.e.* constraints on the overall cardinality of concepts). Prominent examples of such DLs are, *e.g.* Statistical $\mathcal{ALC}$ (Peñaloza & Potyka, 2017), and $\mathcal{ALCSCC}$ with Restricted Cardinality

Boxes (Baader, 2017). In this section we will see how the notion of scattered forward unravellings can be adjusted so that the construction from the previous section additionally preserves *ERCBoxes*, a broad class of linear constraints over the domain.

**Definition 4.15.** *A* semi-restricted cardinality constraint *is an expression $\delta$ of the form*[11]

$$\delta := \mathrm{N}_1 \cdot x_{\mathrm{C}_1} + \ldots + \mathrm{N}_k \cdot x_{\mathrm{C}_k} + M \leq \mathrm{N}_{k+1} \cdot x_{\mathrm{C}_{k+1}} + \ldots + \mathrm{N}_{k+\ell} \cdot x_{\mathrm{C}_{k+\ell}},$$

*where all $\mathrm{C}_i$ are concept names, $x_{\mathrm{C}_i}$ are non-negative integer variables, and all $\mathrm{N}_i$ as well as $M$ are non-negative integer constants, for all $1 \leq i \leq k+\ell$. A solution $\mathfrak{s}$ for $\delta$ is a mapping of variables to non-negative integers under which $\delta$ evaluates to true. An interpretation $\mathcal{I}$ satisfies $\delta$ (written $\mathcal{I} \models \delta$) whenever the mapping $x_{\mathrm{C}} \mapsto |\mathrm{C}^{\mathcal{I}}|$ is a solution for $\delta$.*

*An* extended restricted cardinality box (ERCBox) *(Baader et al., 2020) is a positive boolean combination of semi-restricted cardinality constraints. The notion of solutions and satisfaction by interpretations is lifted to ERCBoxes in the obvious way.*

An important property of ERCBoxes is that they enjoy arbitrarily large solutions:

**Lemma 4.16.** *If an ERCBox $\mathcal{E}$ has a solution $\mathfrak{s}$, then for any positive integer $n$, the mapping $\mathfrak{s}' : x \mapsto n \cdot \mathfrak{s}(x)$ is also a solution for $\mathcal{E}$. Thus solvable ERCBoxes have arbitrarily large solutions.*

*Proof sketch.* Routine calculations. □

Having a finite model $\mathcal{I}$ of an ERCBox $\mathcal{E}$, it can happen that none of the scattered forward unravellings of $\mathcal{I}$ is a model of $\mathcal{E}$, as we did not care about statistical quantities of elements in the constructed models. In order to produce finite models of ERCBoxes, we design a way of "repairing" scattered forward unravellings, in order to restore the satisfaction of ERCBoxes.

Before we move to the main result, we introduce a handy notion of *forward duplication* of domain elements, which is completely independent from the presented unravellings (and thus can be potentially useful for future applications). The key idea is to make a copy of an input element and connect it to all neighbours of the original.

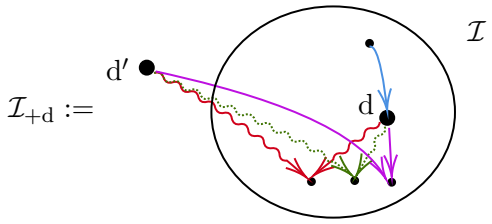
**Definition 4.17.** *Let $\mathcal{I}$ be an interpretation and let $\mathrm{d} \in \Delta^{\mathcal{I}}$ be a domain element. The* d-forward-duplication *of $\mathcal{I}$ is an interpretation $\mathcal{I}_{+\mathrm{d}}$ defined as follows:*

- $\Delta^{\mathcal{I}_{+\mathrm{d}}} := \Delta^{\mathcal{I}} \cup \{\mathrm{d}'\}$ *for a fresh element* $\mathrm{d}'$ *(where $\cup$ denotes disjoint union).*
- $\mathcal{I}_{+\mathrm{d}}$ *restricted to $\Delta^{\mathcal{I}}$ is isomorphic to $\mathcal{I}$.*
- *For each concept name $\mathrm{C} \in \mathbf{N_C}$ we have $\mathrm{d}' \in \mathrm{C}^{\mathcal{I}_{+\mathrm{d}}}$ iff $\mathrm{d} \in \mathrm{C}^{\mathcal{I}}$.*
- *For each role name $r \in \mathbf{N_R}$ and all $\mathrm{e} \in \Delta^{\mathcal{I}}$ we have $(\mathrm{d}', \mathrm{e}) \in r^{\mathcal{I}_{+\mathrm{d}}}$ iff $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$.*

*We will call* $\mathrm{d}'$ *a* copy *of* d. *We stress that* $\mathrm{d}'$ *has no "incoming edges" even if* d *has.*

11. Note the asymmetry in the definition: we do not allow for "spare" integer constants on the RHS. This is because we do not want to give cardinality constraints the power to express nominals.

The notion of d-forward-duplication can be visualised as follows:




The process of duplication can be repeated multiple times, which we formalise next. Given an interpretation $\mathcal{I}$ and finite set $S := \{(\mathrm{d}_1, n_1), (\mathrm{d}_2, n_2), \ldots, (\mathrm{d}_k, n_k)\} \subseteq 2^{(\Delta^{\mathcal{I}} \times \mathbb{N}_+)}$, the $S$-forward-duplication of $\mathcal{I}$ is an interpretation obtained by iterative application of $\mathrm{d}_i$-forward-duplication $n_i$ times for each $1 \leq i \leq k$. For readers eager to see the definition:

**Definition 4.18.** *Let $\mathcal{I}$ be an interpretation and take $S := \{(\mathrm{d}_1, n_1), (\mathrm{d}_2, n_2), \ldots, (\mathrm{d}_k, n_k)\} \subseteq 2^{(\Delta^{\mathcal{I}} \times \mathbb{N}_+)}$. The* $S$-forward-duplication *of $\mathcal{I}$ is an interpretation $\mathcal{I}_{+S}$ defined as follows:*

- $\Delta^{\mathcal{I}_{+S}} := \Delta^{\mathcal{I}} \cup \{\mathrm{d}_i^{(j_i)} \mid 1 \leq i \leq k, 1 \leq j_i \leq n_i\}$, *where all the elements $\mathrm{d}_i^{(j)}$ are fresh.*
- *$\mathcal{I}_{+S}$ restricted to $\Delta^{\mathcal{I}}$ and $\mathcal{I}$ are isomorphic.*
- *For each concept name $\mathrm{C} \in \mathbf{N_C}$ we have $\mathrm{d}_i^{(j)} \in \mathrm{C}^{\mathcal{I}_{+S}}$ iff $\mathrm{d}_i \in \mathrm{C}^{\mathcal{I}}$.*
- *For each role name $r \in \mathbf{N_R}$ and all $\mathrm{e} \in \Delta^{\mathcal{I}}$ we have $(\mathrm{d}_i^{(j)}, \mathrm{e}) \in r^{\mathcal{I}_{+S}}$ iff $(\mathrm{d}_i, \mathrm{e}) \in r^{\mathcal{I}}$.*

The key property of duplication, which is immediate from the definition, is as follows:

**Fact 4.19.** *Let $\mathcal{I}$ be a* finite *interpretation. Then for any* finite *$S := \{(\mathrm{d}_1, n_1), \ldots, (\mathrm{d}_k, n_k)\} \subseteq 2^{(\Delta^{\mathcal{I}} \times \mathbb{N}_+)}$ and any concept name $\mathrm{A} \in \mathbf{N_C}$, the following equation holds:*

$$|\mathrm{A}^{\mathcal{I}_{+S}}| = |\mathrm{A}^{\mathcal{I}}| + \sum_{i \in \{1,2,\ldots,k\},\ \mathrm{d}_i \in \mathrm{A}^{\mathcal{I}}} n_i$$

With a suitable notion of duplication at hand, we will first see that duplication can be used to restore satisfaction of ERCBoxes by scattered unravellings, and second, that all the good properties of scattered forward unravellings are preserved. More precisely we will show:

**Lemma 4.20.** *Let $\mathcal{I}$ be a finite model of an ERCBox $\mathcal{E}$. Then for every positive $n \in \mathbb{N}$ and every finite set of names $\mathsf{N}$, there exists a finite set $S \subseteq 2^{(\Delta^{\mathcal{I}} \times \mathbb{N}_+)}$ for which $(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S} \models \mathcal{E}$.*

*Proof sketch.* By introducing fresh concept names $\mathrm{C}_\mathrm{d}$ per each $\mathrm{d} \in \Delta^{\mathcal{I}}$ and interpreting them by $\mathcal{I}$ as singletons $\{\mathrm{d}\}$, we rewrite an input ERCBox into an ERCBox $\mathcal{E}'$ employing the equations

$$x_\mathrm{C} = \sum_{\mathrm{d} \in \mathrm{C}^{\mathcal{I}}} x_{\mathrm{C}_\mathrm{d}}.$$

Having the solution $\mathfrak{s}$ for $\mathcal{E}$ based on cardinalities of concepts in $\mathcal{I}$, we take a large solution $\mathfrak{s}'$ for $\mathcal{E}'$ (obtained by multiplying $\mathfrak{s}$), which assigns positive values that are greater than the size of the domain of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ (existence guaranteed by Lemma 4.16). We conclude by duplicating $\mathrm{d} \in \Delta^{\mathcal{I}}$ with $S$ sufficiently many times in order to fulfill the equation $|\mathrm{C}_\mathrm{d}^{(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}}| = \mathfrak{s}'(\mathrm{C}_\mathrm{d})$. □

For the preservation of good properties of scattered unravellings treated by duplication, we are going to show that the analogue of Corollary 4.12 and Property 4.14 hold.

**Lemma 4.21.** *Assume $(n, \mathsf{N}, \mathcal{I})$ as in Definition 4.5, and take any finite $S \subseteq 2^{(\Delta^{\mathcal{I}} \times \mathbb{N}_+)}$. Then $(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}$ is a finite $(n, \mathsf{N})$-lff-like interpretation, whose every $n$-neighbourhood can be $\mathsf{N}$-homomorphically mapped to $\mathcal{I}$.*

*Proof.* Finiteness of $(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}$ follows from the following three facts: (i) finiteness of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$, (ii) finiteness of $S$, (iii) Fact 4.19. For the rest of the proof, it suffices to employ Lemma 4.11 after observing that the function mapping duplicated elements to their originals (and behaving as the identity function on other elements) is an $\mathsf{N}$-homomorphism from $(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}$ to $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$. □

We are ready to proceed with our usual list of properties.

**Property 4.22.** *Assume the parameters $(n, \mathsf{N}, \mathcal{I}, S)$ as in Lemma 4.21. Then the analogue of Property 4.14 holds, namely:*

*(A) The interpretations $\mathcal{I}$ and $(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}$ restricted to all $\mathsf{N}$-named elements are isomorphic.*

*(B) For any concept name $\mathrm{C}$ we have that $\mathrm{C}^{\mathcal{I}}$ is non-empty iff $\mathrm{C}^{(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}}$ is non-empty. Similarly, for any role name $r$ we have that $r^{\mathcal{I}}$ is non-empty iff $r^{(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}}$ is non-empty.*

*(C) For any $w \in \Delta^{(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}}$ we have $\mathsf{Conc}_{\mathcal{I}}(\mathsf{orig}^{\star}_{\mathcal{I}}(w)) = \mathsf{Conc}_{(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}}(w)$. Also, for all $v$ with $(w, v) \in r^{(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}}$ for some $r \in \mathbf{N_R}$, we have $\mathsf{Rol}_{\mathcal{I}}(\mathsf{orig}^{\star}_{\mathcal{I}}(w), \mathsf{orig}^{\star}_{\mathcal{I}}(v)) = \mathsf{Rol}_{(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}}(w, v)$.*

*(D) For all $w$ from $(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}$ we have that $w$ in $(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}$ and $\mathsf{orig}^{\star}_{\mathcal{I}}(w)$ in $\mathcal{I}$ have the same number of successors satisfying the same roles and concepts,* i.e. *an analogue of Item (D) of Property 4.4 holds.*

*(E) For any $w$ from $(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}$ we have that $w$ and $\mathsf{orig}^{\star}_{\mathcal{I}}(w)$ are directed-path-equivalent,* i.e. *an analogue of Item (E) of Property 4.4 holds.*

*The function $\mathsf{orig}^{\star}_{\mathcal{I}}$ mentioned above is the mapping that first maps all duplicates to their originals (and behaves like identity on other elements) and then employs $\mathsf{orig}_{\mathcal{I}}$.*

*Proof sketch.* It suffices to apply Property 4.14 after noticing that for any element $w$ from $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ and its duplicated copy $w'$ in $(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}$, the sets of successors of $w$ and $w'$ are equal. □

We say that an abstract DL $\mathcal{DL}$ is *preserved under scattered forward unravellings with rebalancing* if for all finitely satisfiable $\mathcal{DL}$-KBs $\mathcal{K}$, their finite models $\mathcal{I}$, and for all but finitely many positive integers $n \in \mathbb{N}$, there is a finite set $S \subseteq 2^{(\Delta^{\mathcal{I}} \times \mathbb{N}_+)}$ for which $(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S} \models \mathcal{K}$.

**Theorem 4.23.** *If an abstract DL $\mathcal{DL}$ is preserved under scattered forward unravellings with rebalancing then $\mathcal{DL} \in \mathscr{C}_{\mathsf{flf}}$.*

*Proof sketch.* The proof is analogous to the proof of Theorem 4.3. □

A direct application of Theorem 4.23 is the exact complexity for the UCQ entailment problem over $\mathcal{ALCSCC}$ ontologies with ERCBoxes (Baader et al., 2020, Section 2 and Definition 6). Let $\mathcal{K} := (\mathcal{A}, \mathcal{T}, \mathcal{E})$ be a satisfiable $\mathcal{ALCSCC}$ knowledge base, composed of an ABox $\mathcal{A}$,

a TBox $\mathcal{T}$, and an ERCBox $\mathcal{E}$. W.l.o.g. we can assume that (by routine renaming à la Scott) that all concepts names appearing in $\mathcal{K}$ are of depth at most one, and that the ERCBox $\mathcal{E}$ is of the form from Definition 4.15 (the original definition is broader). Take a finite model $\mathcal{I} \models \mathcal{K}$ and any positive integer $n$, and let $\mathcal{J}$ be the $(n, \mathsf{ind}(\mathcal{K}))$-scattered-forward-unravelling of $\mathcal{I}$. Applying Lemma 4.20 we infer the existence of a set $S$ for which $\mathcal{J}_{+S} \models \mathcal{E}$. By Item (A) we deduce $\mathcal{J}_{+S} \models \mathcal{A}$ and by Item (D) we deduce $\mathcal{J}_{+S} \models \mathcal{T}$. Thus $\mathcal{J}_{+S} \models \mathcal{K}$. Since $\mathcal{J}_{+S}$ is an $(n, \mathsf{ind}(\mathcal{K}))$-lff-like model that covers $\mathcal{I}$ (consult Lemma 4.21), we conclude:

**Corollary 4.24.** *The description logic $\mathcal{ALCSCC}$ with ERCBoxes belongs to $\mathscr{C}_{\mathsf{flf}}$.*

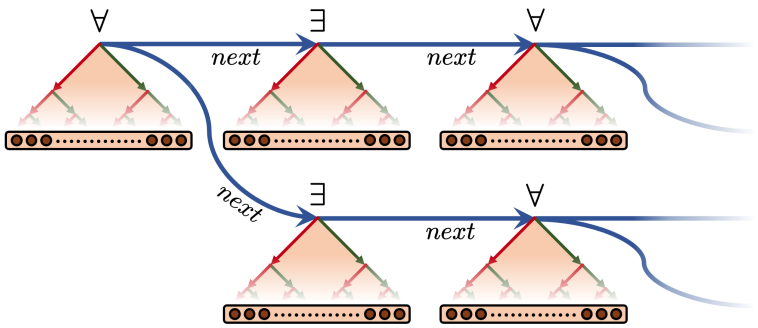
## 5. Query Entailment in the Presence of the Self Operator

As we mentioned in the introduction, the results obtained in the previous sections provide a nearly-complete classification of the complexity of conjunctive query entailment over $\mathcal{ALC}$ extended with popular primitive features. Such a classification excludes the $\mathsf{Self}$ operator, supported by the OWL 2 Web Ontology Language and the DL $\mathcal{SROIQ}$ (Horrocks et al., 2006). The logic $\mathcal{ALC}^{\mathsf{Self}}$ is obtained by extending $\mathcal{ALC}$ with concepts of the form $(\exists r.\mathsf{Self})$ for all $r \in \mathbf{N_R}$. Their interpretation $(\exists r.\mathsf{Self})^{\mathcal{I}}$ is defined simply as $\{\mathrm{d} \mid (\mathrm{d}, \mathrm{d}) \in r^{\mathcal{I}}\}$.

The main result of the forthcoming section is rather surprising: we will show that CQ entailment over $\mathcal{ALC}^{\mathsf{Self}}$ KBs is exponentially harder than CQ entailment over $\mathcal{ALC}$ KBs.

### 5.1 A High-Level Overview of the Encoding

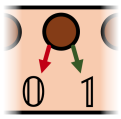
Let $\mathcal{M}$ be an ATM. The core contribution of this section is to present a polynomial-time reduction that, given $\mathcal{M}$, constructs a pair $(\mathcal{K}_{\mathcal{M}}, q_{\mathcal{M}})$ — composed of an $\mathcal{ALC}^{\mathsf{Self}}$ knowledge base and a conjunctive query — such that $\mathcal{K}_{\mathcal{M}} \not\models q_{\mathcal{M}}$ iff $\mathcal{M}$ is accepting. Intuitively, the models of $\mathcal{K}$ will encode accepting quasi-runs of $\mathcal{M}$, *i.e.* trees in which every node is a meaningful configuration of $\mathcal{M}$, but the tape contents of consecutive configurations might not be in sync as they should. The query $q_{\mathcal{M}}$ will be responsible for detecting such errors. Hence, the existence of a countermodel for $\mathcal{K}_{\mathcal{M}}$ and $q_{\mathcal{M}}$ will coincide with the existence of an accepting run of $\mathcal{M}$. The intended models of $\mathcal{K}_{\mathcal{M}}$ look as follows:



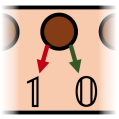

The depicted triangles are called the *configuration trees* and encode configurations of $\mathcal{M}$. The information contained in these configuration trees is "superimposed" on identical *configuration units*: full binary trees of height $\mathtt{N}{+}1$ decorated with many self-loops[12] that will

12. The concrete purpose of the abundant presence of self-loops will only become clear later, starting from Corollary 5.5.

provide the "navigational infrastructure" for the query $q_{\mathcal{M}}$ to detect "tape mismatches". Every such tree has $2^{\mathtt{N}}$ nodes at its $\mathtt{N}$-th level and each of these nodes represents a single tape cell of a machine. However, somehow unexpectedly, we do not just label them directly with concepts representing a letter from the alphabet. Instead, every node at the $\mathtt{N}$-th level also has two children labelled (from left to right) respectively with either $\mathbb{0}$ and $\mathbb{1}$, or with $\mathbb{1}$ and $\mathbb{0}$. Whenever the left child is in $\mathbb{0}$ and the right child is in $\mathbb{1}$, we think that their parent represents a cell filled with the letter $\mathtt{0}$, while the converse situation encodes a cell filled with $\mathtt{1}$.



encoding of tape cell carrying symbol $\mathtt{0}$



encoding of tape cell carrying symbol $\mathtt{1}$

This encoding will be useful to avoid a seemingly required disjunction in the construction of $q_{\mathcal{M}}$. Lastly, the roots of configuration units store all remaining necessary information required for encoding: the current state of $\mathcal{M}$, the previous and the current head position as well as the transition used to arrive at this node from the previous configuration. Finally, the roots of configuration trees are interconnected by the role *next* indicating that $(\mathrm{r}, \mathrm{r}') \in \mathit{next}^{\mathcal{I}}$ holds iff the configuration represented by $\mathrm{r}'$ is a quasi-successor of the configuration of r.

### 5.2 Configuration Units

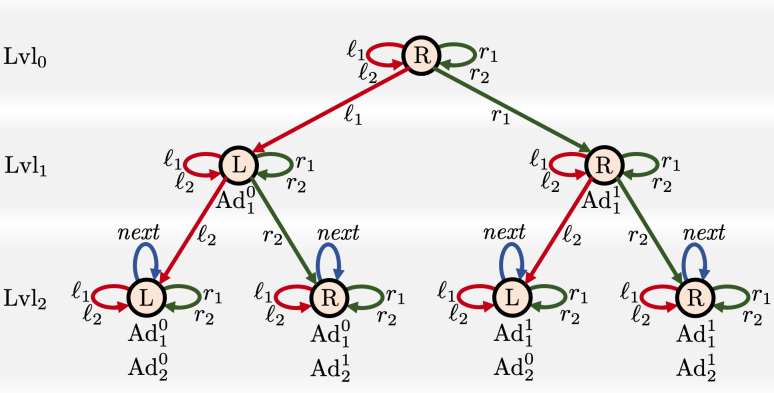
In our encoding, a vital role is played by *n-configuration units*, which will later form the backbone of configuration trees. Roughly speaking, each $n$-configuration unit is a full binary tree of depth $n$, decorated with certain concepts, roles, and self-loops. We introduce configuration units by providing the formal definition, followed by a graphical depiction and an intuitive description. In order to represent configuration units inside interpretations, we employ role names from $\mathbf{R}_{\mathit{unit}}$ as well as concept names from $\mathbf{C}_{\mathit{unit}}$:

$$\mathbf{R}_{\mathit{unit}} := \{\ell_i, r_i, \mathit{next} \mid 1 \leq i \leq n\}, \qquad \mathbf{C}_{\mathit{unit}} := \{\mathrm{Lvl}_0, \mathrm{Lvl}_i, \mathrm{L}, \mathrm{R}, \mathrm{Ad}_i^0, \mathrm{Ad}_i^1 \mid 1 \leq i \leq n\}.$$

**Definition 5.1** (configuration unit)**.** *Given a positive integer $n$, an* $n$-configuration unit $\mathcal{U}$ *is an interpretation $(\Delta^{\mathcal{U}}, \cdot^{\mathcal{U}})$ fulfilling all the conditions below:*

- $\Delta^{\mathcal{U}} = \{0,1\}^{\leq n} := \{w \in \{0,1\}^* \mid |w| \leq n\}$,
- $\mathrm{L}^{\mathcal{U}} \setminus \{\varepsilon\} = \{w0 \in \Delta^{\mathcal{U}}\}$, $\mathrm{R}^{\mathcal{U}} = \Delta^{\mathcal{U}} \setminus \mathrm{L}^{\mathcal{U}}$,
- $\mathrm{Lvl}_i^{\mathcal{U}} = \{w \in \Delta^{\mathcal{U}} \mid |w| = i\}$, $\mathit{next}^{\mathcal{U}} = \{(w,w) \mid |w| = n\}$,
- $\ell_i^{\mathcal{U}} = \{(w, w0) \mid |w| = i{-}1\} \cup \{(w,w) \mid w \in \Delta^{\mathcal{U}}\}$,
- $r_i^{\mathcal{U}} = \{(w, w1) \mid |w| = i{-}1\} \cup \{(w,w) \mid w \in \Delta^{\mathcal{U}}\}$,
- $(\mathrm{Ad}_i^b)^{\mathcal{U}} = \{w \in \Delta^{\mathcal{U}} \mid |w| \geq i$ *and its $i$-th letter is* $b\}$.

The following drawing depicts a 2-configuration unit.




As one can see, the nodes in the tree are layered into levels according to their distance from the root. Nodes at the $i$-th level are members of the $\mathrm{Lvl}_i$ concept and their distance from the root is equal to $i$. Next, each non-leaf node at the $i$-th level has two children, the left one and the right one (satisfying, respectively, the concepts L and R) and is connected to them via the role $\ell_i$ and $r_i$, respectively. All nodes are equipped with $\ell_i$- and $r_i$-self-loops and all leaves are additionally endowed with *next*-loops. With all nodes inside the tree, we naturally associate their addresses, *i.e.* their "numbers" when nodes from the $i$-th level are enumerated from left to right. In order to encode the address of a given node at the $i$-th level, we employ concepts $\mathrm{Ad}_1^b, \mathrm{Ad}_2^b, \ldots, \mathrm{Ad}_i^b$ with "values" $b$ either 0 or 1, meaning that a node is in $\mathrm{Ad}_j^b$ iff the $j$-th bit of its address is equal to $b$. The most significant bit is $\mathrm{Ad}_1^b$.

We next proceed with an axiomatisation of $n$-configuration units in $\mathcal{ALC}^{\mathsf{Self}}$, obtained with the forthcoming GCIs. As usual in such encodings, we cannot formalise such structures up to isomorphism, but the axiomatisation provided is sufficient in a sense made formally precise in the subsequent lemmas.

1. Each node is at exactly one level.

   **(LvlCov)** $\top \sqsubseteq \bigsqcup_{i=0}^{n} \mathrm{Lvl}_i$

   **(LvlDisj[i,j])** $\mathrm{Lvl}_i \sqcap \mathrm{Lvl}_j \sqsubseteq \bot$ (with $0 \leq i < j \leq n$)

2. All nodes carry self-loops for all role names from $\mathbf{R}_{unit}$ except *next* and all leaf nodes (and only they) carry a *next*-loop.

   **(all-loops-but-next)** $\top \sqsubseteq \bigsqcap_{s \in \mathbf{R}_{unit} \setminus \{next\}} \exists s.\mathsf{Self}$

   **(leaves-next-loop)** $\mathrm{Lvl}_n \equiv \exists next.\mathsf{Self}$

3. Every node is either a "left" node or a "right" node.

   **(LRCov)** $\top \sqsubseteq \mathrm{L} \sqcup \mathrm{R}$ **(LRDisj)** $\mathrm{L} \sqcap \mathrm{R} \sqsubseteq \bot$

4. Each node at any level $0 \leq i < n$ has two successors (one left and one right).

   **(LsuccLvl[i])** $\mathrm{Lvl}_i \sqsubseteq \exists \ell_{i+1}.(\mathrm{Lvl}_{i+1}) \sqcap \forall \ell_{i+1}.(\mathrm{Lvl}_{i+1} \to \mathrm{L})$

**(RsuccLvl[i])** $\mathrm{Lvl}_i \sqsubseteq \exists r_{i+1}.(\mathrm{Lvl}_{i+1}) \sqcap \forall r_{i+1}.(\mathrm{Lvl}_{i+1} \to \mathrm{R})$

5. Address information for the nodes is created bit-wise and propagated down the tree. That is, once we are in the left (resp. right) node on the $i$-th level, this node and all nodes further below will have the $i$-th bit of their address set to 0 (resp. 1). Below we have $1 \leq i \leq n$, $b \in \{0,1\}$ and $0 \leq j < i$.

**(LBitZero[i])** $\mathrm{Lvl}_i \sqcap \mathrm{L} \sqsubseteq \mathrm{Ad}_i^0$ **(RBitOne[i])** $\mathrm{Lvl}_i \sqcap \mathrm{R} \sqsubseteq \mathrm{Ad}_i^1$
**(AdDisj[i])** $\mathrm{Ad}_i^0 \sqcap \mathrm{Ad}_i^1 \sqsubseteq \bot$ **(AdLvlDisj[i,j])** $\mathrm{Ad}_i^b \sqcap \mathrm{Lvl}_j \sqsubseteq \bot$

**(PropBit[i])** $\mathrm{Ad}_i^b \sqsubseteq \bigsqcap_{j=1}^{n} \forall \ell_j.\mathrm{Ad}_i^b \sqcap \forall r_j.\mathrm{Ad}_i^b$

This finishes the axiomatisation of $n$-configuration units. Let $\mathcal{K}^n_{unit}$ denote the KB composed of all GCIs presented so far. What remains to be done is to show that our axiomatisation is correct, in the sense of the following two lemmas. Their proofs are routine, hence the reader may skip them at first reading.

**Lemma 5.2.** *Each $n$-configuration unit is a model of $\mathcal{K}^n_{unit}$.*

*Proof sketch.* Take any $n$-configuration unit $\mathcal{U}$ and go through all axioms $\alpha$ of $\mathcal{K}^n_{unit}$ showing that $\mathcal{U} \models \alpha$. This is tedious but there is no hidden difficulty there. □

The proof of the next lemma is by constructing an $n$-configuration unit by starting from an element d and recursively traversing $\ell_i$ and $r_i$ roles.

**Lemma 5.3.** *For any model $\mathcal{I}$ of $\mathcal{K}^n_{unit}$ and any $\mathrm{d} \in \mathrm{Lvl}_0^{\mathcal{I}}$ there is an $n$-configuration unit $\mathcal{U}$ and a homomorphism $\mathfrak{h}$ from $\mathcal{U}$ into $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$.*

*Proof sketch.* Let $\mathcal{U}$ be an $n$-configuration unit with $\varepsilon \in \mathrm{L}^{\mathcal{U}}$ iff $\mathrm{d} \in \mathrm{L}^{\mathcal{I}}$, and that interprets of all role and concept names outside $\mathbf{R}_{unit} \cup \mathbf{C}_{unit}$ as empty sets. It is obvious that exactly one such unit exists. We are going to define a function $\mathfrak{h} : \Delta^{\mathcal{U}} \to \Delta^{\mathcal{I}}$ inductively. Denoting the restriction of $\mathfrak{h}$ to $\{0,1\}^{\leq k}$ by $\mathfrak{h}_{\leq k}$, our inductive assumption states, for a given $k \leq n$, that $\mathfrak{h}_{\leq i}$ is defined for all $i < k$ and $\mathfrak{h}_{\leq i}$ is a homomorphism from $\mathcal{U}{\restriction_{\{0,1\}^{\leq k}}}$ to $\mathcal{I}$.

We first set $\mathfrak{h}(\varepsilon) := \mathrm{d}$. For the inductive step, we assume the hypothesis for $1 \leq k \leq n$ and take a word $w \in \{0,1\}^{k-1}$. We are going to define $\mathfrak{h}(w0)$ as follows (the case of $\mathfrak{h}(w1)$ is symmetric). Note that since $\mathfrak{h}(w) \in \mathrm{Lvl}_{k-1}^{\mathcal{I}}$ (by the fact that $\mathfrak{h}_{\leq k-1}$ is a homomorphism) and since $\mathcal{I} \models$ (LsuccLvl[i]) (for $i$ equal to $k-1$) we conclude the existence of $\mathrm{d}' \in \mathrm{Lvl}_k^{\mathcal{I}}$ satisfying $(\mathfrak{h}(w), \mathrm{d}') \in \ell_k^{\mathcal{I}}$. Note that also $\mathrm{d}' \in \mathrm{L}^{\mathcal{I}} \cap (\mathrm{Ad}_k^0)^{\mathcal{I}}$ holds (by $\mathcal{I} \models$ (LsuccLvl[i]) and $\mathcal{I} \models$ (LBitZero[i]) with $i = k$). Thus, we simply let $\mathfrak{h}(w0) := \mathrm{d}'$. Establishing the property that $\mathfrak{h}_{\leq k}$ is an injective homomorphism is routine and shown in appendix. □

At this point, we would like to give the reader some intuitions on why units are decorated with different self-loops. First, we show that their presence can be exploited to navigate top-down through a given unit.

**Lemma 5.4.** *Let $\mathcal{U}$ be an $n$-configuration unit. Then for all $w \in \Delta^{\mathcal{U}}$ we have $(\varepsilon, w) \in {\ell_1}^{\mathcal{U}} \circ {r_1}^{\mathcal{U}} \circ \ldots \circ {\ell_n}^{\mathcal{U}} \circ {r_n}^{\mathcal{U}}$ with "$\circ$" denoting the composition of relations,* i.e. $s^{\mathcal{U}} \circ t^{\mathcal{U}} := \{(\mathrm{c},\mathrm{e}) \mid (\mathrm{c},\mathrm{d}) \in s^{\mathcal{U}}$ *and* $(\mathrm{d},\mathrm{e}) \in t^{\mathcal{U}}$ *for some* $\mathrm{d}\}$.

*Proof sketch.* Use $s_i^{\mathcal{U}}$ as an abbreviation of ${\ell_1}^{\mathcal{U}} \circ {r_1}^{\mathcal{U}} \circ \ldots \circ {\ell_i}^{\mathcal{U}} \circ {r_i}^{\mathcal{U}}$. Employ induction, stating that for all $1 \leq i \leq n$ we have that all words $w$ of length at most $i$ satisfy $(\varepsilon, w) \in s_i^{\mathcal{U}}$. □

We can now conclude that there is a *single* CQ detecting root-leaf pairs in units.

**Corollary 5.5.** *Let $\mathcal{U}$ be an $n$-configuration unit. There is a single conjunctive query $q_{rl}$ with $x_0, x_{2n} \in \mathrm{Var}(q_{rl})$ and of size polynomial in $n$, such that the set $M = \{(\pi(x_0), \pi(x_{2n})) \mid \mathcal{U} \models_\pi q_{rl}\}$ is equal to the set of root-leaf pairs from $\mathcal{U}$,* i.e. $\mathrm{Lvl}_0^{\mathcal{U}} \times \mathrm{Lvl}_n^{\mathcal{U}}$.

*Proof.* Take $q_{rl} := (\mathrm{Lvl}_0?; \ell_1; r_1; \ldots; \ell_n; r_n; \mathrm{Lvl}_n?)(x_0, x_{2n})$ and apply Lemma 5.4. □

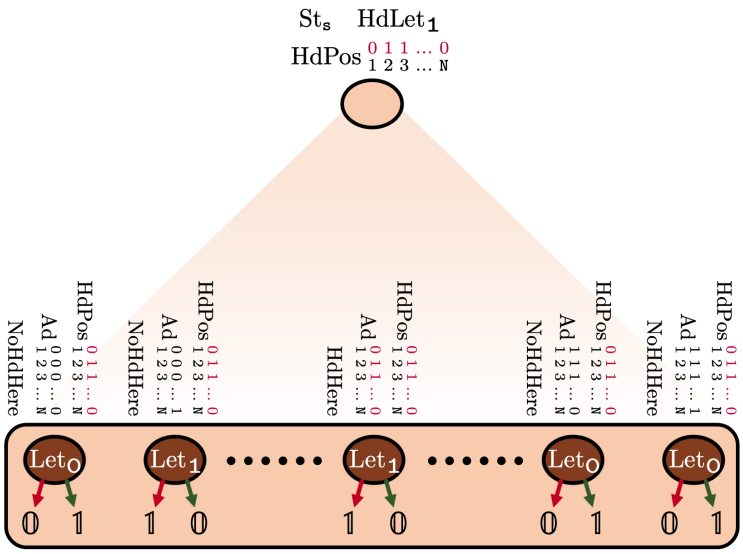
We encourage the reader to play with a query $q := (\mathrm{Lvl}_0?; \ell_1; r_1; \mathrm{Lvl}_1?; \ell_2; r_2; \mathrm{Lvl}_2?)(x_0, x_4)$ and an example 2-configuration unit $\mathcal{U}$ depicted after Definition 5.1. This will make the reader more familiar with the path-syntax of CQs, provide more intuition on the key role played by self-loops in our construction, and justify that indeed any root-leaf pair can be taken as $x_0$ and $x_4$ in an example match $\pi$ witnessing $\mathcal{U} \models_\pi q$.

### 5.3 From Units to Configuration Trees

In the next step, we enrich $(\mathtt{N}{+}1)$-configuration units with additional concepts, allowing the units to represent a meaningful configuration of our ATM $\mathcal{M} = (\mathtt{N}, \mathtt{Q}, \mathtt{Q}_\exists, \mathtt{s}_I, \mathtt{s}_A, \mathtt{s}_R, \mathtt{T})$. To this end, we employ a variety of new concept names from $\mathbf{C}_{conf}$ consisting of

$$\mathbf{C}_{conf} := \{\mathrm{HdHere}, \mathrm{NoHdHere}, \mathrm{St}_{\mathtt{s}}, \mathrm{HdPos}_i^b, \mathrm{HdLet}_{\mathtt{a}}, \mathrm{Let}_{\mathtt{a}}, \mathbb{0}, \mathbb{1} \mid \mathtt{s} \in \mathtt{Q}, b \in \{0,1\}, i \in \{1, \ldots, \mathtt{N}\}, \mathtt{a} \in \{\mathtt{0}, \mathtt{1}\}\}.$$

Before turning to a formal definition we first describe how configurations are structurally represented in models. Recall that a configuration of $\mathcal{M}$ is a word $\mathtt{wsw'}$ with $|\mathtt{ww'}| = 2^{\mathtt{N}}$ (called *tape*) and $\mathtt{s} \in \mathtt{Q}$. In our encoding, this configuration will be represented by an $(\mathtt{N}{+}1)$-configuration unit $\mathcal{C}$ decorated by concepts from $\mathbf{C}_{conf}$. The interpretation $\mathcal{C}$ stores the state $\mathtt{s}$, by associating the state concept $\mathrm{St}_{\mathtt{s}}$ to its root. The tape content $\mathtt{ww'}$ is represented by the internal nodes of $\mathcal{C}$: the $i$-th letter of $\mathtt{ww'}$ (*i.e.* the content of the ATM's $i$-th tape cell) is represented by the $i$-th node (according to their binary addresses) at the $\mathtt{N}$-th level. In case this letter is $\mathtt{0}$, the corresponding node will be assigned the concept $\mathrm{Let}_{\mathtt{0}}$, while $\mathtt{1}$ is represented by $\mathrm{Let}_{\mathtt{1}}$. Yet, for reasons that will become clear only later, the tape cells' content is additionally represented in another way: if it is $\mathtt{0}$, then we label the $i$-th node's left child with $\mathbb{0}$ and its right child with $\mathbb{1}$. The reverse situation is implemented when the node represents the letter $\mathtt{1}$. Finally, there is a unique tape cell that is visited by the head of $\mathcal{M}$ and the node corresponding to this cell is explicitly marked by the concept HdHere while all other "tape cell nodes" are marked by NoHdHere. In order to implement this marking correctly, the head's position's address needs to be explicitly recorded. Consequently, $\mathcal{C}$'s root node stores this address (binarily encoded using the $\mathrm{HdPos}_i^b$ concepts) and from there, these concept assignments are broadcast to and stored in all tape cell nodes on the $\mathtt{N}$-th level. Similarly, we decorate $\mathcal{C}$'s root with the concept $\mathrm{HdLet}_{\mathtt{a}}$ meaning that the current letter scanned by the head is $\mathtt{a}$.




After this informal description and depiction, the formal definition of configuration trees should be plausible.

**Definition 5.6** (configuration tree)**.** *A* configuration tree $\mathcal{C}$ *of* $\mathcal{M}$ *is an interpretation* $\mathcal{C} = (\Delta^{\mathcal{C}}, \cdot^{\mathcal{C}})$ *such that* $\mathcal{C}$ *is an* (N+1)*-configuration unit additionally satisfying:*

- *There exists a unique state* $\mathtt{s} \in \mathtt{Q}$ *such that* $(\mathrm{St}_{\mathtt{s}})^{\mathcal{C}} = \{\varepsilon\}$ *and* $(\mathrm{St}_{\mathtt{s}'})^{\mathcal{C}} = \emptyset$ *for all* $\mathtt{s}' \neq \mathtt{s}$.
- $(\mathrm{Lvl}_{\mathtt{N}+1})^{\mathcal{C}} = \mathbb{0}^{\mathcal{C}} \cup \mathbb{1}^{\mathcal{C}}$ *and* $\mathbb{0}^{\mathcal{C}} \cap \mathbb{1}^{\mathcal{C}} = \emptyset$.
- $(\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}} = \{w \mid w0 \in \mathbb{0}^{\mathcal{C}}, w1 \in \mathbb{1}^{\mathcal{C}}\}$, $(\mathrm{Let}_{\mathtt{1}})^{\mathcal{C}} = \{w \mid w0 \in \mathbb{1}^{\mathcal{C}}, w1 \in \mathbb{0}^{\mathcal{C}}\}$, *and* $(\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}} \cup (\mathrm{Let}_{\mathtt{1}})^{\mathcal{C}} = \mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}}$.
- *There is a* unique *word* $w_{head}$ *of length* N *witnessing* $\mathrm{HdHere}^{\mathcal{C}}{=}\{w_{head}\}$ *and* $\mathrm{NoHdHere}^{\mathcal{C}}{=}\mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}} \setminus \{w_{head}\}$.
- *For* $1 \leq i \leq \mathtt{N}$ *and* $b \in \{0,1\}$ *satisfying* $w_{head} \in (\mathrm{Ad}^b_i)^{\mathcal{C}}$ *we put*[13] $(\mathrm{HdPos}^b_i)^{\mathcal{C}} = \mathrm{Lvl}^{\mathcal{C}}_0 \cup \mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}}$ *and* $(\mathrm{HdPos}^{1-b}_i)^{\mathcal{C}} = \emptyset$.
- $\mathrm{HdLet}^{\mathcal{C}}_{\mathtt{a}} = \{\varepsilon\}$, $\mathrm{HdLet}^{\mathcal{C}}_{\mathtt{1}-\mathtt{a}} = \emptyset$, *where* $\mathtt{a} \in \{\mathtt{0}, \mathtt{1}\}$ *is the unique letter with* $w_{head} \in \mathrm{Let}^{\mathcal{C}}_{\mathtt{a}}$.

We next proceed with the corresponding axiomatisation.

1. To express that $\mathcal{C}$ is an (N+1)-configuration unit we integrate all the GCIs from $\mathcal{K}^{\mathtt{N}+1}_{unit}$.
2. The root of $\mathcal{C}$ is labelled with a unique state concept.

   **(StCov)** $\mathrm{Lvl}_0 \equiv \bigsqcup_{\mathtt{s} \in \mathtt{Q}} \mathrm{St}_{\mathtt{s}}$
   **(StDisj[s,s′])** $\mathrm{St}_{\mathtt{s}} \sqcap \mathrm{St}_{\mathtt{s}'} \sqsubseteq \bot$ (for all $\mathtt{s} \neq \mathtt{s}'$)

3. To axiomatise the coherent representation of the tape's content we employ:

13. This is well-defined since for any $i$, we have that $w_{head}$ belongs to exactly one of $(\mathrm{Ad}^0_i)^{\mathcal{C}}, (\mathrm{Ad}^1_i)^{\mathcal{C}}$ by the definition of a unit.

**(LetDisj)** $\mathbb{0} \sqcap \mathbb{1} \sqsubseteq \bot$ **(LetConDisj)** $\mathrm{Let}_{\mathtt{0}} \sqcap \mathrm{Let}_{\mathtt{1}} \sqsubseteq \bot$

**(LetCov)** $\mathrm{Lvl}_{\mathtt{N}+1} \equiv \mathbb{0} \sqcup \mathbb{1}$ **(LetConCov)** $\mathrm{Let}_{\mathtt{0}} \sqcup \mathrm{Let}_{\mathtt{1}} \equiv \mathrm{Lvl}_{\mathtt{N}}$

**(EncLetZero)** $\mathrm{Let}_{\mathtt{0}} \sqsubseteq \forall \ell_{\mathtt{N}+1}(\mathrm{Lvl}_{\mathtt{N}+1} \to \mathbb{0}) \sqcap \forall r_{\mathtt{N}+1}(\mathrm{Lvl}_{\mathtt{N}+1} \to \mathbb{1})$

**(EncLetOne)** $\mathrm{Let}_{\mathtt{1}} \sqsubseteq \forall \ell_{\mathtt{N}+1}(\mathrm{Lvl}_{\mathtt{N}+1} \to \mathbb{1}) \sqcap \forall r_{\mathtt{N}+1}(\mathrm{Lvl}_{\mathtt{N}+1} \to \mathbb{0})$

4. Next, for the concepts $\mathrm{HdPos}^b_1, \ldots, \mathrm{HdPos}^b_{\mathtt{N}}$ we make sure they encode exactly one proper binary address (meant to encode the head's current position) in the root of $\mathcal{C}$. Below we assume $1 \leq i \leq \mathtt{N}$.

   **(HdPosCov[i])** $\mathrm{Lvl}_0 \sqcup \mathrm{Lvl}_{\mathtt{N}} \equiv \mathrm{HdPos}^0_i \sqcup \mathrm{HdPos}^1_i$

   **(HdPosDisj[i])** $\mathrm{HdPos}^0_i \sqcap \mathrm{HdPos}^1_i \sqsubseteq \bot$

5. Another step is to propagate the head address stored in the root to all nodes on the $\mathtt{N}$-th level of $\mathcal{C}$. Here we exploit the presence of self-loops and Lemma 5.4, and use the following GCIs (for $1 \leq i \leq \mathtt{N}$ and $b \in \{0,1\}$):[14]

   **(PropHdPos[i,b])** $\mathrm{Lvl}_0 \sqcap \mathrm{HdPos}^b_i \sqsubseteq \forall \ell_1 \forall r_1 \ldots \forall \ell_{\mathtt{N}} \forall r_{\mathtt{N}} \, (\mathrm{Lvl}_{\mathtt{N}} \to \mathrm{HdPos}^b_i)$

6. We distinguish between the node representing the cell visited by the head (assigning HdHere) and the other cell nodes (assigning NoHdHere) by having every cell node compare their address (stored in the $\mathrm{Ad}^b_i$ concepts) with the head address received through the broadcast from the root.

   **(HdHereCov)** $\mathrm{HdHere} \sqcup \mathrm{NoHdHere} \equiv \mathrm{Lvl}_{\mathtt{N}}$

   **(HdHereEqualAdr)** $\mathrm{Lvl}_{\mathtt{N}} \sqcap \bigsqcap_{i=1}^{\mathtt{N}} \bigsqcup_{b \in \{0,1\}} (\mathrm{Ad}^b_i \sqcap \mathrm{HdPos}^b_i) \sqsubseteq \mathrm{HdHere}$

   **(NoHdHereDiffrAdr)** $\mathrm{Lvl}_{\mathtt{N}} \sqcap \bigsqcup_{i=1}^{\mathtt{N}} \bigsqcup_{b \in \{0,1\}} (\mathrm{Ad}^b_i \sqcap \mathrm{HdPos}^{1-b}_i) \sqsubseteq \mathrm{NoHdHere}$

7. We synchronise the letter scanned by the head of $\mathcal{M}$ with its "recording" in the root (below $\mathtt{a} \in \{\mathtt{0}, \mathtt{1}\}$).

   **(HdLetCov)** $\mathrm{HdLet}_{\mathtt{0}} \sqcup \mathrm{HdLet}_{\mathtt{1}} \equiv \mathrm{Lvl}_0$

   **(RetrHdLet[a])** $\mathrm{Lvl}_0 \sqcap \exists \ell_1 \exists r_1 \ldots \exists \ell_{\mathtt{N}} \exists r_{\mathtt{N}} (\mathrm{HdHere} \sqcap \mathrm{Let}_{\mathtt{a}}) \sqsubseteq \mathrm{HdLet}_{\mathtt{a}}$

   **(HdLetUnique[a])** $\mathrm{Lvl}_0 \sqcap \mathrm{HdLet}_{\mathtt{a}} \sqsubseteq \forall \ell_1 \forall r_1 \ldots \forall \ell_{\mathtt{N}} \forall r_{\mathtt{N}} (\mathrm{HdHere} \to \mathrm{Let}_{\mathtt{a}})$

This finishes the axiomatisation of configuration trees.

For the knowledge base $\mathcal{K}_{conf}$, composed of all presented GCIs, we present its correctness in the following lemmas. Similarly to the previous section, both of them are routine and the reader may omit them at first reading.

**Lemma 5.7.** *Any configuration tree $\mathcal{C}$ is a model of $\mathcal{K}_{conf}$.*

*Proof sketch.* Take a configuration tree and show that it satisfies all the axioms of $\mathcal{K}_{conf}$. □

14. We note that the same can be achieved without exploitation of self-loops by iteratively propagating the $\mathrm{HdPos}^b_i$ through the tree, but the first author believes that the presented formulation is elegant and makes the reader get used to the presence of self-loops.

**Lemma 5.8.** *For any model $\mathcal{I}$ of $\mathcal{K}_{conf}$ and any $\mathrm{d} \in \mathrm{Lvl}_0^{\mathcal{I}}$ there is a configuration tree $\mathcal{C}$ and a homomorphism $\mathfrak{h}$ from $\mathcal{C}$ into $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$.*

*Proof sketch.* By Lemma 5.3 there is an (N+1)-configuration unit $\mathcal{U}$ and a homomorphism $\mathfrak{h} : \mathcal{U} \to \mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$. It suffices to interpret fresh concept symbols from $\mathbf{C}_{conf}$, according to $\mathfrak{h}$, in order to "upgrade" $\mathcal{U}$ to a configuration tree. The rest of the proof is routine. □

### 5.4 Enriching Configuration Trees

Recall that the purpose of configuration trees is to place them inside a model that describes the run of the Turing machine $\mathcal{M}$. In particular, this will require to describe the progression of one configuration to another. In order to prepare for that, we next introduce an extended version of configuration trees that are enriched by additional information pertaining to their predecessor configuration in a run. To this end, we use new concept names from

$$\mathbf{C}_{enr} := \{\mathrm{PTrns}_{\mathtt{t}}, \mathrm{Ini}, \mathrm{PHdHere}, \mathrm{NoPHdHere}, \mathrm{PHdAbv}, \mathrm{NoPHdAbv}, \mathrm{PHdPos}_i^b, \mathrm{PHdLet}_{\mathtt{a}}\},$$

with $\mathtt{t} \in \mathtt{T}$, $1 \leq i \leq \mathtt{N}$, $b \in \{0, 1\}$, and $\mathtt{a} \in \{\mathtt{0}, \mathtt{1}\}$. We use $\mathbf{C}_{ptr}$ to denote $\{\mathrm{Ini}, \mathrm{PTrns}_{\mathtt{t}} \mid \mathtt{t} \in \mathtt{T}\}$.

The concept $\mathrm{PTrns}_{\mathtt{t}}$, assigned to the root, indicates the transition, through which the configuration has been reached from the previous configuration, while Ini is used as its replacement for the initial configuration. In addition, concepts $\mathrm{PHdPos}_i^b$ and $\mathrm{PHdLet}_{\mathtt{a}}$ are attached to the root in order to — in a way very similar to $\mathrm{HdPos}_i^b$ and $\mathrm{HdLet}_{\mathtt{a}}$ — indicate the previous configuration's head position as well as the letter stored in that position *on the current configuration's tape*. For the sake of our encoding we also employ the concepts PHdHere, NoPHdHere that play the role analogous to the HdHere and NoHdHere concepts from configuration-trees.[15] For technical reasons, we also introduce the concepts PHdAbv and NoPHdAbv that will label nodes on the (N+1)-th level iff their parent is labelled with the corresponding concept from {PHdHere, NoPHdHere}.

We proceed with the formal definition of the structures just described.

**Definition 5.9** (enriched configuration tree)**.** *An* enriched configuration tree $\mathcal{E}$ *of* $\mathcal{M}$ *is an interpretation $\mathcal{E} = (\Delta^{\mathcal{E}}, \cdot^{\mathcal{E}})$ such that $\mathcal{E}$ is a configuration tree additionally satisfying the following conditions on concepts from $\mathbf{C}_{enr}$:*

- *There is exactly one concept $\mathrm{C} \in \mathbf{C}_{ptr}$ for which $\mathrm{C}^{\mathcal{E}} = \{\varepsilon\}$ and for all $\mathrm{C}' \in \mathbf{C}_{ptr} \setminus \{\mathrm{C}\}$ we have $(\mathrm{C}')^{\mathcal{E}} = \emptyset$.*
- $\mathrm{PTrns}^{\mathcal{E}}_{(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',d)} = \{\varepsilon\}$ *implies* $(\mathrm{St}_{\mathtt{s}'})^{\mathcal{E}} = \{\varepsilon\}$ *for all transitions* $(\mathtt{s}, \mathtt{a}, \mathtt{b}, \mathtt{s}', d) \in \mathtt{T}$.
- $\mathrm{PHdHere}^{\mathcal{E}} = \{w_{phd}\}$ *and* $\mathrm{NoPHdHere}^{\mathcal{E}} = \mathrm{Lvl}^{\mathcal{E}}_{\mathtt{N}} \setminus \{w_{phd}\}$ *for the* N*-digit binary word $w_{phd}$ encoding*[16]
  - *the number obtained as $w_{head}$−d (see: Def. 5.6) whenever* $\mathrm{PTrns}^{\mathcal{E}}_{(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',d)} = \{\varepsilon\}$*, or*
  - *the number* 0 *in case* $\mathrm{Ini}^{\mathcal{E}} = \{\varepsilon\}$.

15. For simplicity of axiomatisation, the initial configuration will also carry previous head information, but it will be irrelevant.

16. Here we use the fact that $\mathcal{M}$ never attempts to move left (resp. right) on the left-most (resp. right-most) tape cell.

- $\mathrm{PHdAbv}^{\mathcal{E}} = \{w0, w1 \mid w \in \mathrm{PHdHere}^{\mathcal{E}}\}$ *and* $\mathrm{NoPHdAbv}^{\mathcal{E}} = \mathrm{Lvl}^{\mathcal{E}}_{\mathtt{N}+1} \setminus \mathrm{PHdAbv}^{\mathcal{E}}$.
- $(\mathrm{PHdPos}^b_i)^{\mathcal{E}} = \mathrm{Lvl}^{\mathcal{E}}_0 \cup \mathrm{Lvl}^{\mathcal{E}}_{\mathtt{N}}$ *and* $(\mathrm{PHdPos}^{1-b}_i)^{\mathcal{E}} = \emptyset$ *for all* $1 \leq i \leq \mathtt{N}$ *and* $0 \leq b \leq 1$ *with* $w_{phd} \in (\mathrm{Ad}^b_i)^{\mathcal{E}}$.
- $\mathrm{PHdLet}^{\mathcal{E}}_{\mathtt{a}} = \{\varepsilon\}$ *and* $\mathrm{PHdLet}^{\mathcal{E}}_{\mathtt{1-a}} = \emptyset$, *where* $\mathtt{a}$ *is the unique letter from* $\{\mathtt{0}, \mathtt{1}\}$ *such that* $w_{phd} \in \mathrm{Let}^{\mathcal{E}}_{\mathtt{a}}$.
- $\mathrm{Ini}^{\mathcal{E}} = \{\varepsilon\}$ *implies* $\varepsilon \in \mathrm{L}^{\mathcal{E}}$, $\mathrm{St}^{\mathcal{E}}_{\mathtt{s}_I} = \{\varepsilon\}$, $\mathrm{Let}^{\mathcal{E}}_{\mathtt{0}} = \mathrm{Lvl}^{\mathcal{E}}_{\mathtt{N}}$, *and* $\mathrm{HdPos}^0_i = \mathrm{PHdPos}^0_i = \mathrm{Lvl}^{\mathcal{E}}_0 \cup \mathrm{Lvl}^{\mathcal{E}}_{\mathtt{N}}$ *for all* $1 \leq i \leq \mathtt{N}$.

As usual, we supplement the above definition with the corresponding axiomatisation.

1. We ensure, that the root unambiguously indicates the previous transition (or initiality). Below $\mathtt{t} \neq \mathtt{t}' \in \mathtt{T}$.

$$\textbf{(TrCov)}\ \ \mathrm{Lvl}_0 \ \equiv\ \mathrm{Ini} \sqcup \bigsqcup\nolimits_{\mathtt{t} \in \mathtt{T}} \mathrm{PTrns}_{\mathtt{t}},$$

$$\textbf{(TrInitDisj[t])}\ \ \mathrm{Ini} \sqcap \mathrm{PTrns}_{\mathtt{t}} \sqsubseteq \bot, \qquad \textbf{(TrDisj[t, t'])}\ \ \mathrm{PTrns}_{\mathtt{t}} \sqcap \mathrm{PTrns}_{\mathtt{t}'} \sqsubseteq \bot.$$

2. We provide the encoding of the previous head position and the previous letter scanned by the head. This is achieved by means of the concepts $\mathrm{PHdPos}^b_i$, $\mathrm{PHdLet}_{\mathtt{a}}$, PHdHere, and NoPHdHere in analogy to how it was done for the current head position (see the last four points of the axiomatisation from the previous section). Below we assume $1 \leq i \leq \mathtt{N}$, $b \in \{0, 1\}$, and $\mathtt{a} \in \{\mathtt{0}, \mathtt{1}\}$.

$$\begin{aligned}
&\textbf{(PHdPosCov[i])} && \mathrm{Lvl}_0 \sqcup \mathrm{Lvl}_{\mathtt{N}} \equiv \mathrm{PHdPos}^0_i \sqcup \mathrm{PHdPos}^1_i,\\
&\textbf{(PHdPosDisj[i])} && \mathrm{PHdPos}^0_i \sqcap \mathrm{PHdPos}^1_i \sqsubseteq \bot,\\
&\textbf{(PropPHdPos[i,b])} && \mathrm{Lvl}_0 \sqcap \mathrm{PHdPos}^b_i \sqsubseteq \forall \ell_1 \forall r_1 \ldots \forall \ell_{\mathtt{N}} \forall r_{\mathtt{N}}\, (\mathrm{Lvl}_{\mathtt{N}} \to \mathrm{PHdPos}^b_i),\\
&\textbf{(PHdHereCov)} && \mathrm{PHdHere} \sqcup \mathrm{NoPHdHere} \equiv \mathrm{Lvl}_{\mathtt{N}}\\
&\textbf{(PHdHereEqualAdr)} && \mathrm{Lvl}_{\mathtt{N}} \sqcap \textstyle\bigsqcap_{i=1}^{\mathtt{N}} \bigsqcup_{b \in \{0,1\}} (\mathrm{Ad}^b_i \sqcap \mathrm{PHdPos}^b_i) \ \sqsubseteq \mathrm{PHdHere},\\
&\textbf{(NoPHdHereDiffAdr)} && \mathrm{Lvl}_{\mathtt{N}} \sqcap \textstyle\bigsqcup_{i=1}^{\mathtt{N}} \bigsqcup_{b \in \{0,1\}} (\mathrm{Ad}^b_i \sqcap \mathrm{PHdPos}^{1-b}_i) \sqsubseteq \mathrm{NoPHdHere},\\
&\textbf{(PHdLetCov)} && \mathrm{PHdLet}_{\mathtt{0}} \sqcup \mathrm{PHdLet}_{\mathtt{1}} \equiv \mathrm{Lvl}_0,\\
&\textbf{(RetrPHdLet[a])} && \mathrm{Lvl}_0 \sqcap \exists \ell_1 \exists r_1 \ldots \exists \ell_{\mathtt{N}} \exists r_{\mathtt{N}} (\mathrm{PHdHere} \sqcap \mathrm{Let}_{\mathtt{a}}) \sqsubseteq \mathrm{PHdLet}_{\mathtt{a}},\\
&\textbf{(PHdLetUnique[a])} && \mathrm{Lvl}_0 \sqcap \mathrm{PHdLet}_{\mathtt{a}} \sqsubseteq \forall \ell_1 \forall r_1 \ldots \forall \ell_{\mathtt{N}} \forall r_{\mathtt{N}} (\mathrm{PHdHere} \to \mathrm{Let}_{\mathtt{a}}).
\end{aligned}$$

3. Next, the concepts PHdAbv and NoPHdAbv are assigned via

$$\begin{aligned}
&\textbf{(PHdAbvCov)} && \mathrm{PHdAbv} \sqcup \mathrm{NoPHdAbv} \equiv \mathrm{Lvl}_{\mathtt{N}+1},\\
&\textbf{(PHdAbvDisj)} && \mathrm{PHdAbv} \sqcap \mathrm{NoPHdAbv} \sqsubseteq \bot,\\
&\textbf{(PropPHdAbv)} && \mathrm{PHdHere} \sqsubseteq \forall \ell_{\mathtt{N}+1} \forall r_{\mathtt{N}+1} \mathrm{Lvl}_{\mathtt{N}+1} \to \mathrm{PHdAbv},\\
&\textbf{(PropNoPHdAbv)} && \mathrm{NoPHdHere} \sqsubseteq \forall \ell_{\mathtt{N}+1} \forall r_{\mathtt{N}+1} \mathrm{Lvl}_{\mathtt{N}+1} \to \mathrm{NoPHdAbv}.
\end{aligned}$$

4. We ensure consistency of the current configuration with the previous transition. Below we assume that $(\mathtt{s}, \mathtt{a}, \mathtt{b}, \mathtt{s}', d) \in \mathtt{T}$.

$$\textbf{(TransiCons)}\ \ \mathrm{PTrns}_{(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',d)} \sqsubseteq \mathrm{PHdLet}_{\mathtt{b}} \sqcap \mathrm{St}_{\mathtt{s}'} \sqcap \text{“}\mathrm{PHdPos} + d = \mathrm{HdPos}\text{”},$$

where the last right-hand-side expression, specifying decrements or increments of binary encodings of numbers, is implemented in a usual way (Baader et al., 2017, p. 127) via:

$$\bigsqcup_{i=1}^{\mathtt{N}}\Big(\mathrm{A}_i^0 \sqcap \mathrm{B}_i^1 \sqcap \prod_{j=1}^{i-1}(\mathrm{A}_j^1 \sqcap \mathrm{B}_j^0) \sqcap \prod_{j=i+1}^{\mathtt{N}}((\mathrm{A}_j^1 \sqcap \mathrm{B}_j^1) \sqcup (\mathrm{A}_j^0 \sqcap \mathrm{B}_j^0))\Big)$$

with A := PHdPos and B := HdPos if $d = +1$, and with A and B swapped if $d = -1$.

5. We encode the initial configuration as follows.

**(IC)** $\mathrm{Ini} \sqsubseteq \mathrm{Lvl}_0 \sqcap \mathrm{L} \sqcap \mathrm{St}_{\mathtt{s}_I} \sqcap \prod_{i=1}^{\mathtt{N}}(\mathrm{HdPos}_i^0 \sqcap \mathrm{PHdPos}_i^0) \sqcap \forall \ell_1 \forall r_1 \ldots \forall \ell_{\mathtt{N}} \forall r_{\mathtt{N}}(\mathrm{Lvl}_{\mathtt{N}} \rightarrow \mathrm{Let}_{\mathtt{O}})$,

For the KB $\mathcal{K}_{enr}$, composed of all the GCIs presented so far, we show correctness in the following lemmas. The proofs are routine and similar to the proofs from the previous section.

**Lemma 5.10.** *Any enriched configuration tree of $\mathcal{E}$ is a model of $\mathcal{K}_{enr}$.*

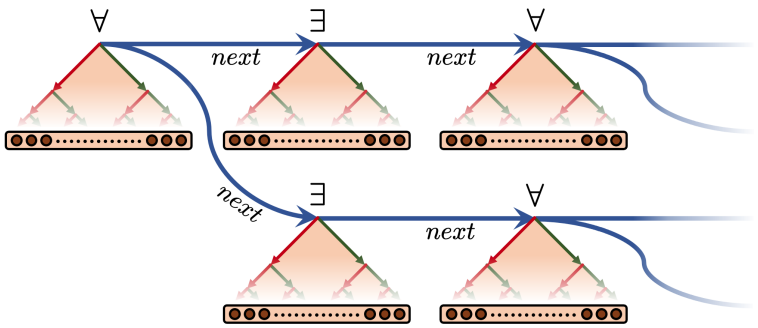
*Proof sketch.* Similarly to the previous proofs, take any enriched configuration tree $\mathcal{E}$ and exhaustively prove that it satisfies all axioms of $\mathcal{K}_{enr}$. There is no hidden difficulty here. □

**Lemma 5.11.** *For any model $\mathcal{I}$ of $\mathcal{K}_{enr}$ and any $\mathrm{d} \in \mathrm{Lvl}_0^{\mathcal{I}}$, there is an enriched configuration tree $\mathcal{E}$ and a homomorphism $\mathfrak{h}$ from $\mathcal{E}$ into $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$.*

*Proof sketch.* We follow the proof scheme of Lemma 5.8. We take a homomorphism $\mathfrak{h}$ from a configuration tree $\mathcal{C}$ to $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$, guaranteed by Lemma 5.8, and then by interpreting fresh concept names from $\mathbf{C}_{enr}$ we "upgrade" $\mathcal{C}$ to an enriched configuration tree. □

### 5.5 Describing Accepting Quasi-Runs

Recall that a quasi-run $\mathfrak{R}$ of $\mathcal{M}$ is simply a tree labelled with configurations of $\mathcal{M}$ where the root is labelled with the initial configuration $\mathtt{s}_I\mathtt{O}^{2^{\mathtt{N}}}$. Each node representing an existential configuration has one child labelled with a quasi-successor configuration, while each node representing a universal configuration has two children labelled by quasi-successor configurations obtained via different transitions.




In order to represent an accepting quasi-run by a model, we employ the notion of a *quasi-computation tree* $\mathcal{Q}$, a structure intuitively defined from some $\mathfrak{R}$ as follows: replace every node of $\mathfrak{R}$ by its corresponding configuration tree, adequately enriched with information about its generating transition and the predecessor configuration. The roots of these enriched

configuration trees are linked via the *next* role to express the quasi-succession relation of $\mathfrak{R}$. The roots of enriched configuration trees representing universal configurations are chosen to be labelled with L, their left *next*-child with L and their right *next*-child with R (both corresponding to existential configurations). As expected, the Ini concept decorates the root of the distinguished enriched configuration tree that represents $\mathfrak{R}$'s initial configuration. As our attention is restricted to *accepting* quasi-runs $\mathfrak{R}$, we require that no enriched configuration tree occurring in $\mathcal{Q}$ carries a rejecting state. We now give a formal definition of such a $\mathcal{Q}$.

**Definition 5.12** (quasi-computation tree)**.** *A* quasi-computation tree $\mathcal{Q}$ *of* $\mathcal{M}$ *is an interpretation* $\mathcal{Q} = (\Delta^{\mathcal{Q}}, \cdot^{\mathcal{Q}})$ *satisfying the following properties:*

- $\Delta^{\mathcal{Q}} := \mathfrak{T} \times \{0,1\}^{\leq \mathtt{N}+1}$*, where* $\mathfrak{T}$ *is*[17] *a prefix-closed subset of* $\{\mathfrak{1o}, \mathfrak{oo}\}^* \cdot \{\varepsilon, \mathfrak{o}, \mathfrak{1}\}$ *with* $\mathfrak{w1} \in \mathfrak{T}$ *implying* $\mathfrak{wo} \in \mathfrak{T}$*.*
- *For every* $\mathfrak{w} \in \mathfrak{T}$*, the substructure of* $\mathcal{Q}$ *induced by* $\{\mathfrak{w}\} \times \{0,1\}^{\leq \mathtt{N}+1}$ *is isomorphic to an enriched configuration tree of* $\mathcal{M}$ *via the isomorphism* $(\mathfrak{w}, w) \mapsto w$*.*
- $(\varepsilon, \mathfrak{w}) \in \mathrm{R}^{\mathcal{Q}}$ *if* $\mathfrak{w}$ *ends with* $\mathfrak{1}$*, otherwise* $(\varepsilon, \mathfrak{w}) \in \mathrm{L}^{\mathcal{Q}}$*.*
- *For any* $\mathfrak{w} \neq \mathfrak{w}'$ *and arbitrary* $w, w' \in \{0,1\}^{\leq \mathtt{N}+1}$ *we have that* $((\mathfrak{w}, w), (\mathfrak{w}', w')) \notin s^{\mathcal{Q}}$ *holds for any* $s \in \mathbf{R}_{unit} \setminus \{next\}$*.*
- $next^{\mathcal{Q}} \setminus \{(\mathrm{d}, \mathrm{d}) \mid \Delta^{\mathcal{Q}} \times \Delta^{\mathcal{Q}}\} = \{((\mathfrak{w}, \varepsilon), (\mathfrak{wb}, \varepsilon)) \mid \mathfrak{wb}, \mathfrak{w} \in \mathfrak{T}, \mathfrak{b} \in \{\mathfrak{o}, \mathfrak{1}\}\}$*.*
- $\mathrm{Ini}^{\mathcal{Q}} = \{(\varepsilon, \varepsilon)\}$*.*
- *For any* $\mathfrak{wo} \in \mathfrak{T}$ *with* $(\mathfrak{w}, \varepsilon) \in \mathrm{St}^{\mathcal{Q}}_{\mathtt{s}}$ *and* $(\mathfrak{w}, \varepsilon) \in \mathrm{Let}^{\mathcal{Q}}_{\mathtt{a}}$
  - *if* $\mathfrak{w1} \in \mathfrak{T}$ *then* $(\mathfrak{wo}, \varepsilon) \in \mathrm{PTrns}^{\mathcal{Q}}_{\mathtt{T}_1(\mathtt{s},\mathtt{a})}$ *and* $(\mathfrak{w1}, \varepsilon) \in \mathrm{PTrns}^{\mathcal{Q}}_{\mathtt{T}_2(\mathtt{s},\mathtt{a})}$*,*
  - *if* $\mathfrak{w1} \notin \mathfrak{T}$ *then* $(\mathfrak{wo}, \varepsilon) \in \mathrm{PTrns}^{\mathcal{Q}}_{\mathtt{T}_1(\mathtt{s},\mathtt{a})}$ *or* $(\mathfrak{wo}, \varepsilon) \in \mathrm{PTrns}^{\mathcal{Q}}_{\mathtt{T}_2(\mathtt{s},\mathtt{a})}$*.*
- *If* $(\mathfrak{w}, w) \in \mathrm{HdHere}^{\mathcal{Q}}$ *and* $\mathfrak{wb} \in \mathfrak{T}$ *then* $(\mathfrak{wb}, w) \in \mathrm{PHdHere}^{\mathcal{Q}}$*.*
- $\mathrm{St}^{\mathcal{Q}}_{\mathtt{s}_R} = \emptyset$ *as well as* $(\mathfrak{w}, \varepsilon) \in \mathrm{St}^{\mathcal{Q}}_{\mathtt{s}_A}$ *if and only if* $\mathfrak{w} \in \mathfrak{T}$ *and* $\mathfrak{wo} \notin \mathfrak{T}$*.*

We move on to provide an appropriate axiomatisation.

1. We incorporate all axioms from $\mathcal{K}_{enr}$ to ensure the indicated substructures correspond to enriched computation trees.
2. Every non-final existential configuration has one successor configuration while every non-final universal configuration has two. Final configurations do not have any successors. Below $\mathtt{s}_e \in \mathtt{Q}_\forall \setminus \{\mathtt{s}_A, \mathtt{s}_R\}$, $\mathtt{s}_e \in \mathtt{Q}_\exists \setminus \{\mathtt{s}_A, \mathtt{s}_R\}$, and $\mathtt{s}_f \in \{\mathtt{s}_A, \mathtt{s}_R\}$.

   **(EConfSucc[$\mathtt{s}_e$])** $\mathrm{St}_{\mathtt{s}_e} \sqsubseteq \exists next.\mathrm{L} \sqcap \exists next.\mathrm{R}$
   **(AConfSucc[$\mathtt{s}_a$])** $\mathrm{St}_{\mathtt{s}_a} \sqsubseteq \exists next.\top \sqcap \forall next.\mathrm{L}$
   **(FinConfSucc[$\mathtt{s}_f$])** $\mathrm{St}_{\mathtt{s}_f} \sqsubseteq \forall next.\bot$
3. To transfer the previous head position to the successor configurations we employ (for $1 \leq i \leq \mathtt{N}, b \in \{0,1\}$):

17. This is just a scary-looking definition of a binary tree in which nodes at the $i$-th level have exactly 2 children if $i$ is even and exactly one child otherwise. We use fraktur letters for quasi-computations.

$$\textbf{(TransHeadPos}[i,b]\textbf{)}\ \ \mathrm{Lvl}_0 \sqcap \mathrm{HdPos}_i^b \sqsubseteq \forall \mathit{next}.\mathrm{PHdPos}_i^b$$

4. For any $\mathtt{s}_\exists \in \mathtt{Q}_\exists$ we specify that the corresponding configuration tree linked via *next*-role is a successor configuration of the current one.

$$\textbf{(TransiExistState)}\ \ \mathrm{St}_{\mathtt{s}_\exists} \sqcap \mathrm{HdLet}_{\mathtt{a}} \sqsubseteq \bigsqcup\nolimits_{\mathtt{t}\in\mathtt{T}(\mathtt{s}_\exists,\mathtt{a})} \forall \mathit{next}.\mathrm{PTrns}_{\mathtt{t}}$$

5. For every universal state $\mathtt{s}_\forall \in \mathtt{Q}_\forall$ and a letter $\mathtt{a}$ currently scanned by the head there are only two possible choices of transitions.

$$\textbf{(TransiUnivStateL)}\ \ \mathrm{St}_{\mathtt{s}_\forall} \sqcap \mathrm{HdLet}_{\mathtt{a}} \sqsubseteq \forall \mathit{next}.(\mathrm{L} \to \mathrm{PTrns}_{\mathtt{T}_1(\mathtt{s}_\forall,\mathtt{a})})$$
$$\textbf{(TransiUnivStateR)}\ \ \mathrm{St}_{\mathtt{s}_\forall} \sqcap \mathrm{HdLet}_{\mathtt{a}} \sqsubseteq \forall \mathit{next}.(\mathrm{R} \to \mathrm{PTrns}_{\mathtt{T}_2(\mathtt{s}_\forall,\mathtt{a})})$$

6. Since we want to have accepting quasi-runs of $\mathcal{M}$ only, we state that we never encounter the rejecting state.

$$\textbf{(NoRejectState)}\ \ \mathrm{St}_{\mathtt{s}_R} \sqsubseteq \bot$$

Let $\mathcal{T}_\mathcal{M}$ be the set of all GCIs presented so far and let $\mathcal{A}_\mathcal{M}$ be an ABox composed of a single axiom $\mathrm{Ini}(\mathtt{a})$ for a fresh individual name $\mathtt{a}$. Put $\mathcal{K}_\mathcal{M} := (\mathcal{A}_\mathcal{M}, \mathcal{T}_\mathcal{M})$. We claim that:

**Lemma 5.13.** *The knowledge base $\mathcal{K}_\mathcal{M}$ is of size polynomial in $|\mathcal{M}|$. Any accepting quasi-computation tree $\mathcal{Q}$ of $\mathcal{M}$ is a model of $\mathcal{K}_\mathcal{M}$.*

*Proof sketch.* The former part follows immediately from the construction, hence we focus on the latter part. To prove $\mathcal{Q} \models \mathcal{K}_{enr}$ notice that (1) by the 2nd item of Definition 5.12 all the substructures of $\mathcal{Q}$ induced by $\{\mathfrak{w}\}\times\{0,1\}^{\leq \mathtt{N}+1}$ are isomorphic to some computation tree and hence, by Lemma 5.10 they satisfy $\mathcal{K}_{enr}$, (2) the use of roles from $\mathbf{R}_{unit} \setminus \{\mathit{next}\}$ is restricted to enriched configuration trees and hence $\mathcal{Q}$ satisfies all the GCIs not involving *next* and (3) the only GCI involving *next* from $\mathcal{K}_{enr}$ is (leaves-next-loop) and it is satisfied in $\mathcal{Q}$ due to the mentioned isomorphism property. Proving the satisfaction of other GCIs is routine. □

**Lemma 5.14.** *For any model $\mathcal{I}$ of $\mathcal{K}_\mathcal{M}$ there exists an accepting quasi-computation tree $\mathcal{Q}$ and a homomorphism $\mathfrak{h} : \mathcal{Q} \to \mathcal{I}$ with $\mathfrak{h}(\varepsilon,\varepsilon) = \mathtt{a}^\mathcal{I}$.*

*Proof.* We construct a tree $\mathfrak{T}$ and its origin function $\mathfrak{f} : \mathfrak{T} \to \mathcal{I}$ as follows. First, let $\varepsilon \in \mathfrak{T}$ and $\mathfrak{f}(\varepsilon) = \mathtt{a}^\mathcal{I}$. We next proceed as follows: take any word $\mathfrak{w} \in \mathfrak{T}$ and consider three cases:

- $\mathfrak{f}(\mathfrak{w})$ is labelled with a non-final universal state. Hence, by the first axiom provided, we know that $\mathfrak{f}(\mathfrak{w})$ has at least two *next*-successors, one of which is in $\mathrm{L}^\mathcal{I}$ and the other in $\mathrm{R}^\mathcal{I}$. Call them, respectively, $\mathrm{e}_l, \mathrm{e}_r$. Hence, we extend $\mathfrak{T}$ with the words $\mathfrak{w}\mathfrak{o}, \mathfrak{w}\mathfrak{1}$ and extend the function $\mathfrak{f}$ with $\mathfrak{f}(\mathfrak{w}\mathfrak{o}) := \mathrm{e}_l$ and $\mathfrak{f}(\mathfrak{w}\mathfrak{1}) := \mathrm{e}_r$. Repeat the process from $\mathfrak{w}\mathfrak{o}$ and $\mathfrak{w}\mathfrak{1}$.
- $\mathfrak{f}(\mathfrak{w})$ is labelled with a non-final existential state. Then we take its *next*-successor e and extend $\mathfrak{T}$ with $\mathfrak{w}\mathfrak{o}$ and $\mathfrak{f}$ with $\mathfrak{f}(\mathfrak{w}\mathfrak{o}) := \mathrm{e}$. Repeat the process from $\mathfrak{w}\mathfrak{o}$.
- $\mathfrak{f}(\mathfrak{w})$ is labelled with a final state. No action required.

We associate a word $\mathfrak{w} \in \mathfrak{T}$ with an enriched configuration tree $\mathcal{E}_\mathfrak{w}$ such that there is a homomorphism $\mathfrak{g}_\mathfrak{w}$ from $\mathcal{E}_\mathfrak{w}$ to $\mathcal{I}$ with $\mathfrak{f}(\mathfrak{w}) = \mathfrak{g}_\mathfrak{w}(\varepsilon)$. The existence of $\mathcal{E}_\mathfrak{w}$ and $\mathfrak{g}_\mathfrak{w}$ is provided by Lemma 5.11. Finally, we decorate each node of $\mathcal{E}_\mathfrak{w}$ with "Pr" concepts as suggested by the homomorphism $\mathfrak{g}_\mathfrak{w}$. A $\mathfrak{T}$-quasi-computation tree $\mathcal{Q}$ is then defined by stipulating that, for every $\mathfrak{w} \in \mathfrak{T}$, the substructure of $\mathcal{Q}$ induced by $\{\mathfrak{w}\} \times \{0,1\}^{\leq \mathtt{N}+1}$ be isomorphic to the decorated $\mathcal{E}_\mathfrak{w}$. The homomorphism $\mathfrak{h} : \Delta^\mathcal{Q} \to \mathcal{I}$ is then defined componentwise by $(\mathfrak{w}, w) \mapsto \mathfrak{g}_\mathfrak{w}(w)$, essentially taking the disjoint unions of the homomorphisms for all enriched configuration trees. Since all the roles except *next* are restricted to the components and we made sure that the roots of $\mathcal{Q}$ were created from the elements linked via *next*-roles, we conclude that $\mathfrak{h}$ is the claimed homomorphism. □

### 5.6 Detecting Faulty Runs with a Single CQ

We finally have reached the point where querying comes into play. Our last goal is to design *one single* conjunctive query that detects "faulty configuration progressions" in quasi-computation trees, meaning that it matches a pair of two positions in consecutive configuration trees representing the same cell and being untouched by the head of $\mathcal{M}$ yet storing different letters. Note that the lack of such cells in a quasi-computation tree means that any two consecutive configuration trees represent not only quasi-successor configuration but actually proper successors and hence the structure as such even represents a "proper" run. We start by formalising our requirements for such a query:

**Lemma 5.15.** *There exists a conjunctive query $q_\mathcal{M}$ of size polynomial in $\mathtt{N}$ with two distinguished variables $x, y$ such that for all quasi-computation trees $\mathcal{Q}$ we have $\mathcal{Q} \models_\pi q_\mathcal{M}$ iff there exists a word $\mathfrak{w}$, a letter $\mathfrak{b}$ and a word $w$ of length $\mathtt{N}{+}1$ such that:*

- $\pi(x) = (\mathfrak{w}, w)$, $\pi(y) = (\mathfrak{w}\mathfrak{b}, w)$,
- $\pi(y) \in \mathrm{NoPHdAbv}^\mathcal{Q}$,
- $\pi(x) \in \mathbb{0}^\mathcal{Q}$ *and* $\pi(y) \in \mathbb{1}^\mathcal{Q}$.

Note the asymmetry in the 3rd bullet point above – we ignore the reverse constellation. Yet, due to our encoding if the reverse situation occurs then so does the original one. Hence, every mismatch in a sense causes two inconsistencies from the point of $\mathtt{N}{+}1$-level nodes. This solves the mystery of introducing level $\mathtt{N}{+}1$ in our configuration trees and the particular encoding of tape symbols: it is crucial for catching faulty progressions by using one single CQ. Before proving Lemma 5.15 we show how it implies the main theorem here, namely:

**Theorem 5.16.** *Conjunctive query entailment over $\mathcal{ALC}^{\mathsf{Self}}$-KBs is 2ExpTime-hard.*

*Proof.* Since co2ExpTime=2ExpTime, it is sufficient to show that CQ non-entailment over $\mathcal{ALC}^{\mathsf{Self}}$ KBs is 2ExpTime-hard. Take $\mathcal{K}_\mathcal{M}$ as defined in Section 5.5 and $q_\mathcal{M}$ as given by Lemma 5.15. Since both $\mathcal{K}_\mathcal{M}$ and $q_\mathcal{M}$ are of size polynomial in $|\mathcal{M}|$, it remains to show that $\mathcal{K}_\mathcal{M} \not\models q_\mathcal{M}$ iff $\mathcal{M}$ is accepting. The "if" direction is easy: we take an accepting run of $\mathcal{M}$ and turn it into a quasi-computation tree $\mathcal{Q}$. By Lemma 5.13 we conclude that $\mathcal{Q} \models \mathcal{K}_\mathcal{M}$. We also have that $\mathcal{Q} \not\models q$ due to the fact that any two consecutive configuration trees represent proper successor configurations. For the second direction it suffices to show that if $\mathcal{M}$ is not accepting then $\mathcal{K}_\mathcal{M} \models q_\mathcal{M}$. Indeed, assume that $\mathcal{M}$ is not accepting and consider an arbitrary

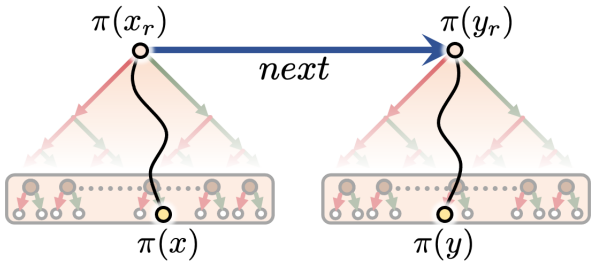
model $\mathcal{I}$ of $\mathcal{K}_\mathcal{M}$ (in case $\mathcal{K}_\mathcal{M}$ is unsatisfiable then trivially $\mathcal{K}_\mathcal{M} \models q_\mathcal{M}$). By Lemma 5.14 there is a quasi-computation tree $\mathcal{Q}$ and a homomorphism $\mathfrak{h} : \mathcal{Q} \to \mathcal{I}$ with $\mathfrak{h}(\varepsilon, \varepsilon) = \mathtt{a}^\mathcal{I}$. But this quasi-computation tree must represent a "faulty" run – in the opposite case it would correspond to an accepting run of $\mathcal{M}$, which does not exist by assumption. Hence there must be a match of $q_\mathcal{M}$ to $\mathcal{Q}$. As query matches are preserved under homomorphisms, we conclude $\mathcal{I} \models q_\mathcal{M}$. Thus all models $\mathcal{I}$ of $\mathcal{K}_\mathcal{M}$ have matches of $q_\mathcal{M}$, which implies $\mathcal{K}_\mathcal{M} \models q_\mathcal{M}$. □

In the forthcoming query definitions, we employ a convenient naming scheme. By writing $q[x, y]$ we indicate that the variables $x, y \in \mathrm{Var}(q)$ are *global* (*i.e.* the same across (sub)queries that we might join together) while its other variables are *local* (*i.e.* pairwise different from any variables occurring in other queries — this can always be enforced by renaming). Going back to the query, we proceed as follows. We first prepare a query $q_{main}[x, y]$ with two global distinguished variables $x, y$ that relates any two domain elements whenever they are leaf nodes of consecutive computation trees. Then $q_{main}[x, y]$ is combined with queries $q^i_{adr}[x, y]$ for all $1 \leq i \leq \mathtt{N}{+}1$ with the intended meaning that $x$ and $y$ have the same $i$-th bit of their addresses. Additionally, our final query will require that $x$ be mapped to a node satisfying $\mathbb{0}$ and $y$ to a node satisfying $\mathbb{1}$ and NoPHdHere.

To construct $q_{main}[x, y]$ we essentially employ Lemma 5.4.

**Lemma 5.17.** *There exists a conjunctive query $q_{main}[x, y]$ of size polynomial in $|\mathcal{M}|$ such that for any quasi-computation tree $\mathcal{Q}$ the set $M_{q_{main}} := \{(\pi(x), \pi(y)) \mid \mathcal{Q} \models_\pi q_{main}\}$ is composed precisely of any pair of leaves of two consecutive configuration trees of $\mathcal{Q}$. Formally:*

$$M_{q_{main}} = \{((\mathfrak{w}, w), (\mathfrak{w}\mathfrak{b}, v)) \in \Delta^\mathcal{Q} \mid |w| = |v| = \mathtt{N}{+}1, \mathfrak{b} \in \{0, 1\}\}.$$




*Proof sketch.* Take $q_{main} := q_{rl}[x_r, x] \wedge next(x_r, y_r) \wedge q_{rl}[y_r, y]$ and employ Corollary 5.5. □

The next part of our query construction focuses on sub-queries $q^i_{adr}[x, y]$ that are meant to relate leaves having equal $i$-th bits of addresses. In order to construct it we combine together several smaller queries, written in path syntax below.

- We let $q_\downarrow[x, y] := (\ell_1; r_1; \ldots; \ell_{\mathtt{N}+1}; r_{\mathtt{N}+1})(x, y)$ define the *top-down query*. It intuitively traverses an enriched configuration tree in a top-down manner. Note that $q_\downarrow[x, y]$ is actually the major sub-query of $q_{rl}[x, y]$.
- The *$\ell_i$-top-down query* $q_{\ell i\downarrow}[x, y]$ is similar to $q_\downarrow[x, y]$, but with the $\ell_i; r_i$ part replaced by just $\ell_i$. The intended behaviour is that again a tree is traversed from root to leaves, but this time, an $\ell_i$ edge must be crossed when going from the $(i-1)$-th to the $i$-th level. The *$r_i$-top-down query* $q_{ri\downarrow}[x, y]$ is defined alike, by replacing $\ell_i; r_i$ in $q_\downarrow[x, y]$ with $r_i$.

An important ingredient in the construction is the query $q^{i\text{-th bit}}_{=0}[x,y]$ defined as follows:

$$\mathrm{Lvl}_{\mathtt{N}+1}(x) \wedge q_{\ell i\downarrow}[x',x] \wedge \mathit{next}(x',y') \wedge q_{\ell i\downarrow}[y',y] \wedge \mathrm{Lvl}_{\mathtt{N}+1}(y).$$

In total analogy, we define $q^{i\text{-th bit}}_{=1}[x,y]$ by using $q_{ri\downarrow}$ instead of $q_{\ell i\downarrow}$. Any match $\pi$ of the query $q^{i\text{-th bit}}_{=b}[x,y]$ instantiates the variables $x$ and $y$ in a quasi-computation tree $\mathcal{Q}$ according to one of the following two scenarios: either $\pi(x)=\pi(y)$ or $\pi(x)$ and $\pi(y)$ are leaves in two consecutive enriched configuration trees inside the quasi-computation tree and both of these leaves have their $i$-th address bit set to $b$. The above intuition meets its formalisation in:

**Lemma 5.18.** *Let $\mathcal{Q}$ be a quasi-computation tree and let $M_{q^{i\text{-th bit}}_{=b}} := \{(\pi(x),\pi(y)) \mid \mathcal{Q} \models_\pi q^{i\text{-th bit}}_{=b}\}$ for $b\in\{0,1\}$. Then $M_{q^{i\text{-th bit}}_{=b}}$ is equal to the union of $M^b_1$ and $M^b_2$ given below:*

$$M^b_1 := \{((\mathfrak{w},w),(\mathfrak{w},w))\}, \qquad M^b_2 := \{((\mathfrak{w},ubv),(\mathfrak{w}\mathfrak{b},u'bv')) \mid |u|=|u'|=i{-}1\}.$$

*Proof.* We prove the statement for $b=0$, the case for $b=1$ is symmetric. First we show $M^0_1 \subseteq M_{q^{i\text{-th bit}}_{=0}}$. This is easy: for any leaf $\mathrm{d}=(\mathfrak{w},w)$ we map all variables of $q^{i\text{-th bit}}_{=0}[x,y]$ into d; this is a match due to the presence of all the self-loops at the leaves. To show $M^0_2 \subseteq M_{q^{i\text{-th bit}}_{=0}}$ we take any $\mathrm{d}:=(\mathfrak{w},w)$ and $\mathrm{e}:=(\mathfrak{w}\mathfrak{b},v)$. Let $\pi$ be a variable assignment that maps $x$ to d, $y$ to e, $x'$ to $(\mathfrak{w},\varepsilon)$, $y'$ to $(\mathfrak{w}\mathfrak{b},\varepsilon)$. The variables of $q_{\ell i\downarrow}[x',x]$ are mapped to $(\mathfrak{w},w_j)$, where $w_j$ is the prefix of $w$ of length $j$ following the path from $(\mathfrak{w},\varepsilon)$ to $(\mathfrak{w},w)$ level-by-level. We stress that $((\mathfrak{w},w_{i-1}),(\mathfrak{w},w_i)) \in \ell_i^{\mathcal{Q}}$ holds, which is crucial for the construction to work and that every $(\mathfrak{w},w_j)$ node has all $\ell$- and $r$-loops. The variables of $q_{\ell i\downarrow}[y',y]$ are mapped analogously. After noticing that $\mathrm{d},\mathrm{e}\in\mathrm{Lvl}^{\mathcal{Q}}_{\mathtt{N}+1}$ and that $(\pi(x'),\pi(y')) \in \mathit{next}^{\mathcal{Q}}$ holds, we conclude that $\pi$ is clearly a match of $q^{i\text{-th bit}}_{=0}[x,y]$ to $\mathcal{Q}$.

Now we focus on showing that $M_{q^{i\text{-th bit}}_{=0}[x,y]} \subseteq M^0_1 \cup M^0_2$. Take any match $\pi$ and note that $x,y$ must be mapped to leaves. For $\pi(x')$ and $\pi(y')$ we consider the two cases:

1. $\pi(x')=\pi(y')$. As the roots do not have *next*-loops, $\pi(x')$ must be a leaf. This implies that all variables of $q_{\ell i\downarrow}[x',x]$ map into a single domain element (otherwise we would not reach a leaf after traversing such a path). Arguing similarly we infer that all variables of $q_{\ell i\downarrow}[y',y]$ are mapped to the same element. Thus $\pi(x)=\pi(y)$ holds.
2. $\pi(x')\neq\pi(y')$. Since all incoming *next* roles from leaves are self-loops, we conclude that $\pi(x')$ is the root of some enriched quasi-computation tree and $\pi(y')$ is the root of some corresponding quasi-successor in $\mathcal{Q}$ (by the definition of $\mathit{next}^{\mathcal{Q}}$). By the satisfaction of $q_{\ell i\downarrow}[x',x]$ we know that there exists a sequence of domain elements contributing to a path from $\pi(x')$ to $\pi(x)$ witnessing its satisfaction. Moreover, note that since the subquery $q_{\ell i\downarrow}[x',x]$ leads from the root to a leaf it implies that we necessarily cross the $\ell_i$ role at the $(i{-}1)$-th level, meaning that the $i$-th bit of the address of $\pi(x)$ is equal to 0. Thus we infer that $\pi(x)\in(\mathrm{Ad}^0_i)^{\mathcal{Q}}$. Reasoning analogously we conclude that $\pi(y)\in(\mathrm{Ad}^0_i)^{\mathcal{Q}}$, which finishes the proof. □

We are now ready to present the query $q^i_{adr}[x,y]$ pairing leaves in consecutive enriched configuration trees with coinciding $i$-th address bit:

$$q^i_{adr}[x,y] := q_{main}[x,y] \wedge q^{i\text{-th bit}}_{=0}[x,z] \wedge q^{i\text{-th bit}}_{=1}[z,y].$$

**Lemma 5.19.** *For any quasi-computation tree $\mathcal{Q}$ we have that $M_{q^i_{adr}} := \{(\pi(x), \pi(y)) \mid \mathcal{Q} \models_\pi q^i_{adr}[x, y]\}$ is composed precisely of the leaf pairs in two consecutive enriched configuration trees of $\mathcal{Q}$ having equal $i$-th bit of address. Formally:*

$$M_{q^i_{adr}} = M_{q_{main}} \cap \Big( (\mathrm{Ad}_i^{0\,\mathcal{Q}} \times \mathrm{Ad}_i^{0\,\mathcal{Q}}) \cup (\mathrm{Ad}_i^{1\,\mathcal{Q}} \times \mathrm{Ad}_i^{1\,\mathcal{Q}}) \Big).$$

*Proof.* By employing the definition of the query, Lemma 5.18 and relational calculus we conclude that $M_{q^i_{adr}} = M_{q_{main}} \cap (M_{q^{i\text{-th bit}}_{=0}} \circ M_{q^{i\text{-th bit}}_{=1}}) = M_{q_{main}} \cap ((M_1^0 \cup M_2^0) \circ (M_1^1 \cup M_2^1)) = M_{q_{main}} \cap (M_1^0 \cup M_2^1 \cup M_2^0) = M_2^1 \cup M_2^0$, which concludes the proof. □

We are finally ready to present our query

$$q_{\mathcal{M}} := \bigwedge_{i=1}^{\mathtt{N}+1} q^i_{adr}[x, y] \wedge \mathrm{NoPHdAbv}(y) \wedge \mathbb{0}(x) \wedge \mathbb{1}(y)$$

by means of which we can conclude with the proof of Lemma 5.15.

*Proof of Lemma 5.15.* Let $q_{\mathcal{M}}$ as defined above and observe that its size is clearly polynomial in $\mathtt{N}$. Note that $q_{\mathcal{M}}$ satisfies our requirements: The 1st item follows from two lemmas: the fact that $x$ and $y$ are mapped to leaves of two consecutive enriched configuration trees follows from Lemma 5.17 and the fact that $x$ and $y$ are mapped to nodes having equal addresses follows from Lemma 5.19 applied for every $1 \leq i \leq \mathtt{N}{+}1$. The 2nd and the 3rd points hold since we supplemented our query with $\mathrm{NoPHdAbv}(y) \wedge \mathbb{0}(x) \wedge \mathbb{1}(y)$. □

Hardness, shown in this section, came as a quite surprise to us. In fact, we spent quite some time trying to include self-loops in the notion of locally-forward interpretations from Section 3, but it turned out that the analogous of Lemma 3.17 is incorrect. The key insight of our proof (and maybe the take-home message from this section and the whole paper) is that the presence of $\mathsf{Self}$ allows us to mimic case distinction over paths (and hence the handling of disjunctive information) through concatenation, by providing the opportunity for one of the two disjuncts to idle by "circling in place". On a last note, our result also holds for plain $\mathcal{ALC}^{\mathsf{Self}}$ TBoxes, since the only ABox assertion $\mathrm{Ini}(\mathsf{a})$ can be replaced by the GCI $\top \sqsubseteq \exists aux.\mathrm{Ini}$ for an auxiliary role name $aux$.

**Corollary 5.20.** *Conjunctive query entailment over $\mathcal{ALC}^{\mathsf{Self}}$-TBoxes is 2ExpTime-hard.*

## 6. Conclusions

In this paper, we investigated a model-theoretic criterion, called (finitely) locally forwardness, for description logics. We proved that for logics enjoying such a property the query entailment problem can be solved by means of exponentially many calls to the satisfiability problem of appropriately constructed knowledge bases, having sizes polynomially bounded in terms of the input. This yields, among other results, ExpTime-completeness of the UCQ entailment problem over $\mu\mathcal{ALCSCC}$ knowledge bases, as well as for $\mathcal{ALCSCC}$ w.r.t. ERCBoxes, filling multiple gaps in the literature with a single proof. To make our technique easier to apply in future research, we also provided sufficient conditions, based on suitable model transformations, that can be used to decide whether a given logic is (finitely) locally forward.

Finally, we demonstrated an important limitation of our technique: the notion of forward trees underpinning our model-theoretic criteria cannot be extended even with self-loops. This is due to a very surprising (at least to us) result shown in the paper, namely that the conjunctive query entailment for $\mathcal{ALC}^{\mathsf{Self}}$ happens to be 2ExpTime-hard.

Our paper closes existing open problems rather than producing new ones. A possible direction for future work is to see whether our notion of lff-like structures is relatively maximal guaranteeing a "tamed complexity" of the query entailment problem. More formally:

**Open Problem 6.1.** *Is there a* natural *extension $\mathcal{DL}$ of $\mathcal{ALC}$ with an ExpTime-complete knowledge base satisfiability problem and ExpTime-complete conjunctive query entailment problem, for which there exists an $\mathcal{DL}$-KB $\mathcal{K}$ with arbitrarily large models, so that for all but finitely many $n$, none of the models of $\mathcal{K}$ with more than $n$ elements is $(n, \mathsf{ind}(\mathcal{K}))$-llf-like"?*

We leave to the reader to decide what the word "natural" means in this context. Moreover, we also stress that possible extensions of lff-like structures, *e.g.* with self-loops at arbitrary nodes or by allowing that some of the parent-to-child roles can be "inverted", are not sufficient to guarantee ExpTime-completeness for the query entailment problem, as witnessed by the negative results for querying $\mathcal{ALC}^{\mathsf{Self}}$ and $\mathcal{ALCI}$ TBoxes.

## Acknowledgements

This work is supported by the ERC Consolidator Grant No. 771779 (DeciGUT).

## Appendix A. Missing definitions of $\mu$ and $\mathcal{SCC}$.

In the two following sections we introduce the DL features ($\mathcal{SCC}$) and ($\mu$).

### A.1 Less-known DL Features: $\mathcal{SCC}$

Herein we introduce the very expressive counting feature ($\mathcal{SCC}$), following Baader et al. (2020). We employ the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic (QFBAPA) to express cardinality constraints.

We start with an introduction of QFBAPA. In the logic QFBAPA, one can build *set terms* by applying boolean operations (intersection $\cap$, union $\cup$, and complement $\cdot^c$) to set variables as well as the constants $\emptyset$ and $\mathcal{U}$. Set terms $s, t$ can then be used to state *set constraints*, which are equality and inclusion constraints of the form $s = t, s \subseteq t$, where $s, t$ are set terms. *Presburger Arithmetic (PA) expressions* are built from integer constants and set cardinalities $|s|$ using addition as well as multiplication with an integer constant. They can be used to form *cardinality constraints* of the form $k = \ell, k < \ell, N \,\mathsf{dvd}\, \ell$, where $k, \ell$ are PA expressions, $N$ is an integer constant, and $\mathsf{dvd}$ stands for divisibility. A *QFBAPA formula* is a boolean combination of set and cardinality constraints using connectives $\wedge, \vee, \neg$.

A *substitution* $\sigma$ assigns a finite set $\sigma(\mathcal{U})$ to $\mathcal{U}$, the empty set to $\emptyset$, and subsets of $\sigma(\mathcal{U})$ to set variables. It is extended to set terms by interpreting the boolean operations $\cap$, $\cup$, and $\cdot^c$ as set intersection, set union, and set complement w.r.t. $\sigma(\mathcal{U})$, respectively. The substitution $\sigma$ satisfies the set constraint $s = t$ ($s \subseteq t$) if $\sigma(s) = \sigma(t)$ ($\sigma(s) \subseteq \sigma(t)$). It is further extended to a mapping from PA expressions to integers by interpreting $|s|$ as the cardinality of the finite set $\sigma(s)$, and addition and multiplication with an integer constant

in the usual way. The substitution $\sigma$ satisfies the cardinality constraint $k = \ell$ if $\sigma(k) = \sigma(\ell)$, $k < \ell$ if $\sigma(k) < \sigma(\ell)$, and $N \,\mathsf{dvd}\, \ell$ if the integer constant $N$ is a divisor of $\sigma(\ell)$. The notion of satisfaction of a boolean combination of set and cardinality constraints is now defined in the obvious way by interpreting $\wedge, \vee, \neg$ as in propositional logic. The substitution $\sigma$ is a *solution* of the QFBAPA formula $\phi$ if it satisfies $\phi$ in this sense. A QFBAPA formula $\phi$ is *satisfiable* if it has a solution. Kuncak and Rinard (2007) proved that the satisfiability problem for QFBAPA formulae is NP-complete.

The DL feature ($\mathcal{SCC}$) introduces concepts of the form $succ(\alpha)$, where $\alpha$ is either a set constraint or a cardinality constraint that uses role names and already defined concepts in place of set variables. The formal semantics is presented next. For a given $\mathrm{d} \in \Delta^{\mathcal{I}}$ the substitution $\tau^{\mathcal{I}}_{\mathrm{d}}$ assigns the finite set $\bigcup_{r \in \mathbf{N_R}} \{\mathrm{e} \mid (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}$ to $\mathcal{U}$, the empty set to $\emptyset$, and the sets $\{\mathrm{e} \mid (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}$ to $r$ and $\mathrm{A}^{\mathcal{I}} \cap \bigcup_{r \in \mathbf{N_R}} \{\mathrm{e} \mid (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}$ to A, where $r \in \mathbf{N_R}$ and $\mathrm{A} \in \mathbf{N_C}$ are viewed as set variables. The interpretation function $\cdot^{\mathcal{I}}$ and the substitutions $\tau^{\mathcal{I}}_{\mathrm{d}}$ for $\mathrm{d} \in \Delta^{\mathcal{I}}$ are inductively extended to concepts by interpreting the boolean operators $\sqcap, \sqcup, \neg$ in the usual way and the successor expressions *succ* as follows:

- $succ(\alpha)^{\mathcal{I}} = \{\mathrm{d} \in \Delta^{\mathcal{I}} \mid \text{the substitution } \tau^{\mathcal{I}}_{\mathrm{d}} \text{ satisfies } \alpha\}$,
- $\tau^{\mathcal{I}}_{\mathrm{d}}(succ(\alpha)) = succ(\alpha)^{\mathcal{I}} \cap \bigcup_{r \in \mathbf{N_R}} \{\mathrm{e} \mid (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}$.

This concludes the definition of ($\mathcal{SCC}$). We would also like to point out that a somehow cumbersome definition of ($\mathcal{SCC}$) based on QFBAPA, can be presented equivalently in terms of Presburger Arithmetics as presented by Demri and Lugiez (2010).

### A.2 Less-known DL Features: $\mu$

The next feature ($\mu$) is quite technical. It extends the underlying description logic $\mathcal{DL}$ with fixed-points. For its definition we closely follow the description of $\mu\mathcal{ALCQ}$ by De Giacomo and Lenzerini (1997, Sec. 4).

Let $\mathbf{N_F}$ be a countably infinite set of *fixed-point variables*, that is pairwise disjoint from $\mathbf{N_C}$, $\mathbf{N_R}$, and $\mathbf{N_V}$ . The logic $\mu\mathcal{DL}$ extends the set of concepts constructors of a logic $\mathcal{DL}$ with the use of variables X from $\mathbf{N_F}$ (treated as atomic concepts), and two new "quantified expressions" (called *fixed-point operators*) $\mu$X.C and $\nu$X.C, where C is a concept, with the restriction that only a variable X occurring positively in C can be bound by a fixpoint $\mu/\nu$ in $\mu$X.C and $\nu$X.C. By *positive* we mean that every free occurrence of a variable X is under an even number of negations. A valuation $\eta$ on an interpretation $\mathcal{I}$ is a mapping that assigns variables from $\mathbf{N_F}$ to subsets of $\Delta^{\mathcal{I}}$. For a given valuation $\eta$, we use $\eta[\mathrm{X/E}]$ to denote the valuation identical to $\eta$ with the exception of $\eta[\mathrm{X/E}](\mathrm{X}) \coloneqq \mathrm{E}$.

Take an interpretation $\mathcal{I}$ and a valuation $\eta$. We define the semantics of concepts by associating to $\mathcal{I}$ and $\eta$ an extension function $\cdot^{\mathcal{I}}_{\eta}$ mapping concept to subsets of $\Delta^{\mathcal{I}}$ as follows:

$$\begin{aligned} \mathrm{X}^{\mathcal{I}}_{\eta} &\coloneqq \eta(\mathrm{X}) \text{ for all variables X}, \\ (\mu\mathrm{X.C})^{\mathcal{I}}_{\eta} &\coloneqq \bigcap \Big\{\mathrm{E} \subseteq \Delta^{\mathcal{I}} \mid \mathrm{C}^{\mathcal{I}}_{\eta[\mathrm{X/E}]} \subseteq \mathrm{E}\Big\}, \\ (\nu\mathrm{X.C})^{\mathcal{I}}_{\eta} &\coloneqq \bigcup \Big\{\mathrm{E} \subseteq \Delta^{\mathcal{I}} \mid \mathrm{E} \subseteq \mathrm{C}^{\mathcal{I}}_{\eta[\mathrm{X/E}]}\Big\}, \end{aligned}$$

and with all concepts without variables and fixed-point operators interpreted as usual. The notion of GCIs $\mathrm{C} \sqsubseteq \mathrm{D}$ (where both C and D do not contain free variables) is lifted to the

case with valuation function in a natural way, by requiring that $\mathcal{I} \models \mathrm{C} \sqsubseteq \mathrm{D}$ if and only if for all valuation functions $\eta$ we have $(\mathcal{I}, \eta) \models \mathrm{C}^{\mathcal{I}}_{\eta} \subseteq \mathrm{D}^{\mathcal{I}}_{\eta}$. The notion of knowledge bases and their satisfaction is defined analogously.

## Appendix B. Appendix to Section 3

### B.1 Proof of Property 3.5

*Proof.* Let $\mathcal{I}$ be a countermodel for $\mathcal{K}$ and $q$. By the fact that $\mathcal{K}$ is lff-coverable we infer the existence of a $(|q|, \mathsf{ind}(\mathcal{K}))$-lff-like model for $\mathcal{K}$ and $q$ that covers $\mathcal{I}$. We claim that $\mathcal{J}$ is the desired lff-like countermodel $\mathcal{J}$ for $\mathcal{K}$ and $q$. Indeed, if we would have $\mathcal{J} \models q$ then $\mathcal{J} \models_{\pi} q_i$ (for some $1 \leq i \leq m$ and a match $\pi$). Then the connected components of $\mathcal{J}\restriction_{\{\pi(x) \mid x \in \mathrm{Var}(q_i)\}}$ are of size $\leq |q|$ and hence, can be homomorphically mapped to $\mathcal{I}$ by assumption. This implies $\mathcal{I} \models q_i$, and therefore $\mathcal{I} \models q$, contradicting the countermodelhood of $\mathcal{I}$. □

### B.2 Proof of Lemma 3.8

*Proof.* We first consider the case when $x$ is $\prec$-maximal. Note that the domain of $\mathcal{I}_q^{[x \preceq]}$ then consists of a single element $x$.

- From $\mathrm{d} \in (\mathrm{Subt}^x_q)^{\mathcal{I}}$ to the existence of a homomorphism $\mathfrak{h} : \mathcal{I}_q^{[x \preceq]} \to \mathcal{I}$ defined as $\mathfrak{h}(x) := \mathrm{d}$. It suffices to show that $\mathfrak{h}$ preserves concepts. Let $\mathrm{A} \in \mathbf{N_C}$ be such that $x \in \mathrm{A}^{\mathcal{I}_q^{[x \preceq]}}$. From the definition of $\mathcal{I}^x_q$, we know that $\mathrm{A}(x) \in q$. By the first case of Definition 3.7, the concept A appears as one of the conjuncts of $\mathrm{Subt}^x_q$. Thus, from $\mathrm{d} \in (\mathrm{Subt}^x_q)^{\mathcal{I}}$, we infer $\mathfrak{h}(x){=}\mathrm{d} \in \mathrm{A}^{\mathcal{I}}$. Thus $\mathfrak{h}$ is indeed a homomorphism.

- From the existence of $\mathfrak{h} : \mathcal{I}^x_q \to \mathcal{I}$, defined as $\mathfrak{h}(x) := \mathrm{d}$, to $\mathrm{d} \in (\mathrm{Subt}^x_q)^{\mathcal{I}}$.

  We simply need to take any conjunct from $\mathrm{Subt}^x_q$ and prove the membership of d in it. Here the only option is that the conjunct is of the form $\mathrm{A} \in \mathbf{N_C}$, so by the first case of Definition 3.7, we know that $\mathrm{A}(x) \in q$. By the definition of $\mathcal{I}^x_q$, we get $x \in \mathrm{A}^{\mathcal{I}_q^{[x \preceq]}}$. By applying the fact that $\mathfrak{h}$ is a homomorphism, we infer $\mathfrak{h}(x) = \mathrm{d} \in \mathrm{A}^{\mathcal{I}}$. Thus d belongs to each conjunct of $\mathrm{Subt}^x_q$, resulting in $\mathrm{d} \in (\mathrm{Subt}^x_q)^{\mathcal{I}}$.

Assume that $x$ is not $\prec$-maximal and that for all $y$ with $x \prec y$ the lemma is known to be true. There are two cases:

- From $\mathrm{d} \in (\mathrm{Subt}^x_q)^{\mathcal{I}}$ to the existence of a homomorphism $\mathfrak{h} : \mathcal{I}^x_q \to \mathcal{I}$ with $\mathfrak{h}(x) = \mathrm{d}$.

  From the fact that $\mathrm{d} \in (\mathrm{Subt}^x_q)^{\mathcal{I}}$ and from the last conjunct from Definition 3.7, we obtain that for each variable $y \in \mathsf{Chlds}(x)$ there is a domain element $\mathrm{d}_y \in \Delta^{\mathcal{I}}$ satisfying:

$$(\mathrm{d}, \mathrm{d}_y) \in \left( \bigcap_{r(x,y) \in q} r^{\mathcal{I}} \right) \text{ and } \mathrm{d}_y \in (\mathrm{Subt}^y_q)^{\mathcal{I}}. \qquad (\clubsuit)$$

  Moreover, by the induction hypothesis, for every $y \in \mathsf{Chlds}(x)$ there is a homomorphism $\mathfrak{h}_y : (\mathrm{Subt}^y_q)^{\mathcal{I}} \to \mathcal{I}$ with $\mathfrak{h}_y(y) = \mathrm{d}_y$. Let us define a function $\mathfrak{h} : \mathcal{I}_q^{[x \preceq]} \to \mathcal{I}$ as: $\mathfrak{h}(x) := \mathrm{d}$ and for all $z \in \mathrm{Var}(q)$ we set $\mathfrak{h}(z) := \mathfrak{h}_y(z)$, where $y \in \mathsf{Chlds}(x)$ such that $y \preceq z$. We first

explain why the definition of $\mathfrak{h}$ is correct and then why $\mathfrak{h}$ is indeed a homomorphism, finishing the proof. The correctness of the definition comes directly from the fact that $\mathcal{I}_q^{[x\preceq]}$ is a forward tree: any variable is then either the root of $\mathcal{I}_q^{[x\preceq]}$ or has the unique ancestor being a child of the root. Now, to argue that $\mathfrak{h}$ is a homomorphism, we prove the preservation of (atomic) concepts and roles by $\mathfrak{h}$. To see that $\mathfrak{h}$ preserves concepts, we either (a) use the same reasoning as in the base case for the root variable or (b) invoke an inductive hypothesis that $\mathfrak{h}_y$ are homomorphisms for other variables. For the preservation of roles, note that due to forward-tree-shapedness of $\mathcal{I}_q^{[x\preceq]}$ there are only three cases to consider: for any role name $r$ either $r^{\mathcal{I}_q^{[x\preceq]}}$ is empty (thus we do not need to do anything) or contains pairs of the form (i) $(x, y)$ for $y \in \mathsf{Chlds}(x)$, or (ii) $(z, v)$ with some $z \neq x$. We handle each of these cases separately. For (i) we observe that if $(x, y) \in r^{\mathcal{I}_q^x}$ holds for some variable $y \in \mathsf{Chlds}(x)$ then $r(x, y) \in q$ and by Equation (♣) we conclude $(\mathfrak{h}(x), \mathfrak{h}(y)) = (\mathrm{d}, \mathrm{d}_y) \in r^{\mathcal{I}}$. Finally, for the case (ii), we know that there is a unique variable $y$ from $\mathsf{Chlds}(x)$ satisfying $y \preceq z$. Hence, from the fact that $\mathfrak{h}_y$ is a homomorphism, we get $(\mathfrak{h}(z), \mathfrak{h}(v)) \in r^{\mathcal{I}}$, completing the proof that $\mathfrak{h}$ is also a homomorphism.

- From the existence of a homomorphism $\mathfrak{h} : \mathcal{I}_q^{[x\preceq]} \to \mathcal{I}$ with $\mathfrak{h}(x) = \mathrm{d}$, to $\mathrm{d} \in (\mathrm{Subt}_q^x)^{\mathcal{I}}$.

  It suffices to show that d satisfies each conjunct from $\mathrm{Subt}_q^x$. Showing that d belongs to the first conjunct (*i.e.* these representing atoms of the form $\mathrm{A}(x)$) is the same as in the base case. Thus, we focus only on the last conjunct. Let $\exists \left(\cap_{r(x,y)\in q} r\right) \mathrm{Subt}_q^y$ be any of them and set $\mathrm{d}_y = \mathfrak{h}(y)$. Let $\mathfrak{h}_y$ be the restriction of $\mathfrak{h}$ to $\{z \mid y \preceq z\}$ and note that $\mathfrak{h}_y : \mathcal{I}_q^{[y\preceq]} \to \mathcal{I}$ is a homomorphism with $\mathfrak{h}_y(y) = \mathrm{d}_y$. Hence, by the inductive assumption, we deduce $\mathrm{d}_y \in (\mathrm{Subt}_q^y)^{\mathcal{I}}$. It remains to show $(\mathrm{d}, \mathrm{d}_y) \in \cap_{r(x,y)\in q} r^{\mathcal{I}}$. Take any $r(x, y) \in q$. By the definition of $\mathcal{I}_q^x$, we conclude that $(x, y) \in r^{\mathcal{I}_q^{[x\preceq]}}$. Since $\mathfrak{h}$ is a homomorphism, we get $(\mathfrak{h}(x), \mathfrak{h}(y)) = (\mathrm{d}, \mathrm{d}_y) \in r^{\mathcal{I}}$, as required. Thus, $\mathrm{d} \in (\mathrm{Subt}_q^x)^{\mathcal{I}}$.

We have shown the correctness of the implications in both ways, concluding the proof. □

### B.3 Proof of Lemma 3.13

*Proof.* Assume $\mathcal{I} \models q'$. Since $q'$ is a fork rewriting of $q$, there exists a derivation $q{=}q_n \rightsquigarrow_{\mathsf{fe}} q_{n-1} \rightsquigarrow_{\mathsf{fe}} \ldots \rightsquigarrow_{\mathsf{fe}} q_0{=}q'$. Reasoning inductively, it suffices to show that for all indices $0 \leq i < n$ we have that $\mathcal{I} \models q_i$ implies $\mathcal{I} \models q_{i+1}$. Then we conclude the lemma by taking $i \coloneqq n - 1$. Assume $\mathcal{I} \models q_i$, *i.e.* that there is a homomorphism $\mathfrak{h}_i : \mathcal{I}_{q_i} \to \mathcal{I}$. Since $q_{i+1} \rightsquigarrow_{\mathsf{fe}} q_i$ holds, we can find the variables $x, y, z$ such that (i) $\mathrm{Var}(q_i) \setminus \{x, y, z\} = \mathrm{Var}(q_{i+1}) \setminus \{x, y, z\}$ and (ii) $q_i$ was obtained from $q_{i+1}$ by replacing each occurrence of $x$ or $y$ in any atoms with $z$. Hence, let $\mathfrak{f} : \mathcal{I}_{q_{i+1}} \to \mathcal{I}_{q_i}$ be a function satisfying $\mathfrak{f}(x) = \mathfrak{f}(y) = z$ and $\mathfrak{f}(v) = v$ for all other variables. From (i) and (ii) we immediately infer that $\mathfrak{f}$ is a homomorphism. Thus $(\mathfrak{f} \circ \mathfrak{h}_i) : \mathcal{I}_{q_{i+1}} \to \mathcal{I}$ is a homomorphism, establishing $\mathcal{I} \models q_{i+1}$. □

### B.4 Proof of Lemma 3.17 (if)

*Proof.* By Lemma 3.13, it suffices to show $\mathcal{I} \models q'$. We construct a function $\mathfrak{h} : \mathrm{Var}(q') \to \mathcal{I}$ as follows:

- For every root variable $x \in \texttt{Roots}$ we put $\mathfrak{h}(x) := (\texttt{name}(x))^{\mathcal{I}}$.
- Fix an index $1 \leq i \leq n$. By Item (b) of Definition 3.14 we know that $q'\restriction_{\texttt{SubTree}_i}$ is forward-tree-shaped and let $x_i$ be its root. Moreover, by Item (D) of Definition 3.16 there exists an element $\mathrm{d}_i \in \Delta^{\mathcal{I}}$ satisfying:

$$\Big(\texttt{name}(\texttt{root-of}(i))^{\mathcal{I}}, \mathrm{d}_i\Big) \in \left(\bigcap_{r(\texttt{root-of}(i),x_i)\in q'} r^{\mathcal{I}}\right) \quad \text{and} \quad \mathrm{d}_i \in \Big(\mathrm{Match}_{q'\restriction_{\texttt{SubTree}_i}}\Big)^{\mathcal{I}}. \tag{♠}$$

  From the forward-tree-shapedness of $q'\restriction_{\texttt{SubTree}_i}$ and Lemma 3.8 we conclude the existence of a homomorphism $\mathfrak{h}_i$ from $\mathcal{I}_{q'\restriction_{\texttt{SubTree}_i}}$ to $\mathcal{I}$ with $\mathfrak{h}_i(x_i) = \mathrm{d}_i$. Thus we can simply put $\mathfrak{h}(x) := \mathfrak{h}_i(x)$ for all $x \in \texttt{SubTree}_i$.
- Take any connected component $\hat{q}$ of $q'\restriction_{\texttt{Trees}}$, which by Item (a) of Definition 3.14 is forward-tree-shaped. From the compatibility of $\Pi^{\mathsf{N}}_{q'}$ with $\mathcal{I}$ and Item (A) of Definition 3.16 we know that there is an element $\mathrm{d} \in \Delta^{\mathcal{I}}$ satisfying $\mathrm{d} \in (\mathrm{Match}_{\hat{q}})^{\mathcal{I}}$. Invoking Corollary 3.9, we deduce that there exists a homomorphism $\mathfrak{h}_{\hat{q}} : \mathcal{I}_{\hat{q}} \to \mathcal{I}$. Finally, we put $\mathfrak{h}(x) := \mathfrak{h}_{\hat{q}}(x)$ for all $x \in \mathrm{Var}(\hat{q})$.

Note that the definition of $\mathfrak{h}$ is correct, *i.e.* that every argument has a value assigned and that each argument has only one value assigned, since (i) the sets $\texttt{Roots}, \texttt{SubTree}_1, \ldots, \texttt{SubTree}_n$, $\texttt{Trees}$ induce a partition of $\mathrm{Var}(q)$, (ii) all forward-tree-shaped queries from $\texttt{Trees}$ are variable-disjoint and (iii) the employed homomorphism are functions themselves. Hence, it remains to show that $\mathfrak{h}$ is also a homomorphism from $\mathcal{I}_{q'}$ to $\mathcal{I}$. Proving the preservation of atomic concepts by $\mathfrak{h}$ is immediate: for root variables we employ Item (B) of Definition 3.16, while for the other variables we rely on the fact that the result of $\mathfrak{h}$ is then defined via another homomorphism. For the proof of the preservation of roles by $\mathfrak{h}$, we take any pair $(x, y) \in r^{\mathcal{I}_{q'}}$, or equivalently $r(x, y) \in q'$, and we will show that $(\mathfrak{h}(x), \mathfrak{h}(y)) \in r^{\mathcal{I}}$. By Item (c) of Definition 3.14 we know that there are only four cases to consider, depending on the location of $x$ and $y$:

- Both $x$ and $y$ belong to $\texttt{Roots}$.
  Then $(\mathfrak{h}(x), \mathfrak{h}(y)) = \Big(\texttt{name}(x)^{\mathcal{I}}, \texttt{name}(y)^{\mathcal{I}}\Big) \in r^{\mathcal{I}}$ follows from Item (C) of Definition 3.16.
- There exists an index $1 \leq i \leq n$ such that $x, y \in \texttt{SubTree}_i$.
  Then $(x, y) \in r^{\mathcal{I}_{q'\restriction_{\texttt{SubTree}_i}}}$ holds and we get $(\mathfrak{h}(x), \mathfrak{h}(y)) = (\mathfrak{h}_i(x), \mathfrak{h}_i(y)) \in r^{\mathcal{I}}$ since $\mathfrak{h}_i$ is a homomorphism.
- Both $x$ and $y$ belong to $\texttt{Trees}$.
  From $(x, y) \in r^{\mathcal{I}_{q'}}$ we know that $x, y$ are in the same subtree $\hat{q}$ of $\texttt{Trees}$. Thus, $(x, y) \in r^{\mathcal{I}_{\hat{q}}}$ holds and it suffices to apply the fact that $\mathfrak{h}_{\hat{q}}$ is a homomorphism to get $(\mathfrak{h}(x), \mathfrak{h}(y)) = (\mathfrak{h}_{\hat{q}}(x), \mathfrak{h}_{\hat{q}}(y)) \in r^{\mathcal{I}}$.
- The variables $x$ and $y$ are in two different sets.
  First, from Item (c) of Definition 3.14, we know that there is an $i$ such that $x \in \texttt{Roots}$ satisfies $\texttt{root-of}(i) = x$ and $y = x_i \in \texttt{SubTree}_i$ is the root of $q\restriction_{\texttt{SubTree}_i}$. Second, by Equation (♠) we know that $(\mathfrak{h}(x), \mathfrak{h}(y))$ is actually equal to $\Big(\texttt{name}(\texttt{root-of}(i))^{\mathcal{I}}, \mathrm{d}_i\Big)$

for some already-fixed $\mathrm{d}_i \in \Delta^{\mathcal{I}}$. Finally, by applying the first part of Equation (♠), we get $(\mathfrak{h}(x), \mathfrak{h}(y)) = \left(\mathtt{name}(\mathtt{root\text{-}of}(i))^{\mathcal{I}}, \mathrm{d}_i\right)$ belongs to $r^{\mathcal{I}}$, as required.

We have shown that the function $\mathfrak{h}$ is indeed a homomorphism. Thus $\mathcal{I} \models q'$ holds, implying $\mathcal{I} \models q$. □

### B.5 Proof of Lemma 3.17 (only if)

*Proof.* Let $\pi$ be a match witnessing $\mathcal{I} \models_\pi q$. As a preliminary step, we modify $\pi$ and $q$ slightly, to make them more "forest-like". Then we will define an appropriate splitting.

We construct the query $q'$ by exhaustively applying fork elimination on all "forks" $r(x, z), s(y, z)$ with $\pi(x) = \pi(y)$ (where $r, s$ are not necessarily different). Obviously $\mathcal{I} \models q'$ holds (a match $\pi'$ is for $q'$ is essentially the same as $\pi$ modulo changing the set of variables). Now we will modify $\pi'$. Let $\mathcal{I}'$ be any connected substructure of $\mathcal{I}$ induced by $\pi'$ and let $\mathsf{N}' \subseteq \mathsf{N}$ consists of all names given to elements from $\mathcal{I}'$. Moreover, let $V \subseteq \mathrm{Var}(q')$ be the $\subseteq$-maximal set of variables from $q'$ so that the image of $V$ by $\pi'$ is precisely $\mathcal{I}'$. Note that $|\mathcal{I}'| \leq |q|$ and consider two cases:

- If $\mathsf{N}'$ is empty, then by $(|q|, \mathsf{N})$-lff-likeness of $\mathcal{I}$ we know that there is a homomorphism $\mathfrak{f}$ from $\mathcal{I}'$ to some forward-tree-shaped interpretation $\mathcal{I}''$ and a homomorphism $\mathfrak{g}$ from $\mathcal{I}''$ to $\mathcal{I}$. We next redefine $\pi'$ so that it maps any variable $v \in V$ to $\mathfrak{g}(\mathfrak{f}(\pi'(v)))$. This yields the property that the image of $V$ by $\pi'$ constitutes a forward-tree.
- The case when $\mathsf{N}' \neq \emptyset$ is treated similarly, but we use homomorphisms to $\mathsf{N}'$-rooted forward-forests instead.

Next, we define an $\mathsf{N}$-splitting

$$\Pi^{\mathsf{N}}_{q'} := (\mathtt{Roots}, \mathtt{name}, \mathtt{SubTree}_1, \mathtt{SubTree}_2, \ldots, \mathtt{SubTree}_n, \mathtt{root\text{-}of}, \mathtt{Trees}),$$

where the definitions of its components are provided below.

- The set $\mathtt{Roots}$ is composed of all variables $x \in \mathrm{Var}(q')$ for which $\pi'(x)$ is an $\mathsf{N}$-named element of $\mathcal{I}$. For all such variables $x$ we set $\mathtt{name}(x) := \mathsf{a}$ for any corresponding $\mathsf{a} \in \mathsf{N}$.
- The sets $\mathtt{SubTree}_i$, as their name suggests, are defined by taking subtrees connected to the roots. To simplify the definition, we say that a variable $x$ is *dangling from a root* if there exists a variable $x_r \in \mathtt{Roots}$ and an atom $r(x_r, x)$ in $q'$. Let $D$ be the subset-maximal set of variables from $(\mathrm{Var}(q') \setminus \mathtt{Roots})$ dangling from roots. Take $n := |D|$ and fix an ordering $x_1, x_2, \ldots, x_n$ on the elements from $D$. For any index $1 \leq i \leq n$ we define $\mathtt{SubTree}_i$ as the set composed of $x_i$ and all variables reachable from $x_i$ via a directed path of positive length in the query structure $\mathcal{I}_{q'\restriction_{\mathrm{Var}(q')\setminus\mathtt{Roots}}}$. Observe that $\mathcal{I}\restriction_{\{\pi'(v) \mid v \in \mathtt{SubTree}_i\}}$ is a forward-tree. This follows from the fact that the $|q|$-neighbourhood of $\pi'(x_i)$ in $\mathcal{I}$ is either a forward-tree (and hence we are done) or it is an $\mathsf{N}'$-rooted forward-forest (then since $\pi'(x_i)$ is not $\mathsf{N}$-named it is an inner node of the forest and hence the nodes reachable from it constitute a forward-tree).

  Thus, due to the fact that we eliminated all forks, the underlying query $q'\restriction_{\mathtt{SubTree}_i}$ is forward-tree-shaped.

- We put $\texttt{root-of}(i) \coloneqq x_r^i$, where $x_r^i$ is the root from which $x_i \in D$ is dangling.
  Note that $x_r^i$ is uniquely determined due to the construction of $q'$. Indeed, ad absurdum assume that there is $y_r^i \neq x_r^i$ such that $r(x_r^i, x_i)$ and $s(y_r^i, x_i)$ holds. There are two cases: either $\pi'(x_r^i) = \pi'(y_r^i)$ or $\pi'(x_r^i) \neq \pi'(y_r^i)$. The former case is clearly not possible due to the fact that such "forks" were eliminated in $q'$. In the latter case it implies that there are two $\mathsf{N}$-named elements of $\mathcal{I}$ pointing at $\pi'(x_i)$. Recall that $\mathcal{I}$ is a $(|q|, \mathsf{N})$-locally-forward-forest-like and hence, the local neighbourhood of $\pi'(x_i)$ is a forward-forest with at least two roots, $\pi'(x_r^i)$ and $\pi'(y_r^i)$. Thus the latter case is only possible when $\pi'(x_i)$ is also $\mathsf{N}$-named, but it is not because $x_i \notin \texttt{Roots}$. A contradiction.
- The set $\texttt{Trees}$ contains all other variables from $\mathrm{Var}(q')$.

Now we show that $\Pi_{q'}^{\mathsf{N}}$ is indeed a splitting. We have already argued that $\texttt{name}$ and $\texttt{root-of}$ are functions and that Item (b) and Item (d) of Definition 3.14 hold. It remains to prove that the selected sets induce a partition of $\mathrm{Var}(q')$ as well as the satisfaction of Items (a) and (c) of Definition 3.14.

We start with the former issue. First, note that by the above definitions the set $\texttt{Trees}$ guarantees that all the set components of $\Pi_{q'}^{\mathsf{N}}$ sums to $\mathrm{Var}(q')$ and that $\texttt{Trees}$ are disjoint from the other sets. Moreover, since the variables from $\texttt{Roots}$ were excluded while defining $\texttt{SubTree}_i$ we conclude that $\texttt{Roots} \cap \texttt{SubTree}_i = \emptyset$ for any index $i$. Hence, it suffices to take any two indices $i < j$ and show the disjointness of $\texttt{SubTree}_i$ and $\texttt{SubTree}_j$. Assume towards a contradiction that $\texttt{SubTree}_i \cap \texttt{SubTree}_j \neq \emptyset$. Thus there is a variable $v$ reachable from both $x_i$ and $x_j$ (the roots of $q{\restriction_{\texttt{SubTree}_i}}$ and $q{\restriction_{\texttt{SubTree}_j}}$, different by definition) via directed paths in $\mathcal{I}_{q'{\restriction_{\texttt{SubTree}_i}}}$ and $\mathcal{I}_{q'{\restriction_{\texttt{SubTree}_j}}}$. We consider the following cases:

1. $\texttt{SubTree}_i = \{x_i\}$ and $\texttt{SubTree}_j = \{x_j\}$.
   This implies that $v = x_j$ or $v = x_i$ and contradicts the fact that each of the above sets is a singleton.
2. $v = x_i$ (the case of $v = x_j$ is analogous).
   Hence, we infer the existence of a directed path from $x_j$ to $x_i$ in $\mathcal{I}_{q'{\restriction_{\texttt{SubTree}_j}}}$. Moreover, all the elements on this path are anonymous, since they belong to $\texttt{SubTree}_j$. Note that $x_i$ is also anonymous. Thus there is also a directed path (of positive length!) from $\pi'(x_j)$ to $\pi'(x_i)$ of length $\leq |q'|$ in $\mathcal{I}$. But it contradicts the fact that the $|q|$-neighbourhood of $\pi'(x_i)$ is an $\mathsf{N}'$-forward-forest, with some $\mathsf{N}'$ containing $\texttt{name}(\texttt{root-of}(i))$ and $\texttt{name}(\texttt{root-of}(j))$.
3. $x_i \neq v \neq x_j$.
   Thus there are variables $u \in \texttt{SubTree}_i$, $w \in \texttt{SubTree}_j$ and $z \in \texttt{SubTree}_i \cap \texttt{SubTree}_j$ (with $z$ possibly equal to $v$) such that $r(u, z) \in q'$ and $s(w, z) \in q'$ and $z$ is reachable from both $x_i$ and $x_j$. Since we eliminated all the forks, we know that $\pi'(w) \neq \pi'(u)$. Thus, by applying the fact that $|q|$-neighbourhood of $\pi'(x_i)$ is an $\mathsf{N}'$-forward-forest, we get a contradiction because $\pi'(u), \pi'(w), \pi'(z)$ do not form a forward-forest.

Hence components of $\Pi_{q'}^{\mathsf{N}}$ indeed induce a partition of $\mathrm{Var}(q')$.

Next, we proceed with Item (a). Take any connected component $\hat{q}$ of $q'{\restriction_{\texttt{Trees}}}$. Note that $|\hat{q}| \leq |q|$ and for any variable $v \in \mathrm{Var}(\hat{q})$ we have that $\pi'(v)$ is not $\mathsf{N}$-named. Hence, from the $(|q|, \mathsf{N})$–lff-likeness of $\mathcal{I}$ we infer the substructure induced by $\pi'$ and $\hat{q}$ is a forward-tree, so is $\hat{q}$ (we eliminated all the forks!).

Finally, we need to argue that Item (c) of Definition 3.14 holds. If $x \in \texttt{Roots}$ and $r(x, y) \in q'$ then either if $\pi'(y)$ is $\mathsf{N}$-named then $y \in \texttt{Roots}$ (thus $x, y$ are in the same set) or $y$ is dangling from the root so, by the construction, is in some $\texttt{SubTree}_i$. By construction, $y$ is the root of $q'\restriction_{\texttt{SubTree}_i}$. Otherwise $x \notin \texttt{Roots}$ and we consider the following cases:

1. If $x \in \texttt{SubTree}_i$ and $y \in \texttt{SubTree}_j \cup \texttt{Trees}$ then $y \in \texttt{SubTree}_i$ violating the disjointness of these sets.

2. $y \in \texttt{Roots}$ or ($x \in \texttt{Trees}$ and $y \in \texttt{SubTree}_i$). We get a contradiction with lff-likeness of $\mathcal{I}$.

This finishes the proof that $\Pi^{\mathsf{N}}_{q'}$ is an $\mathsf{N}$-splitting of $q'$. Next, we will argue that $\Pi^{\mathsf{N}}_{q'}$ is compatible with $\mathcal{I}$. Item (A) follows from Corollary 3.9. Items (B) and (C) are immediate by the fact that $\mathcal{I} \models_{\pi'} q'$. Finally, for Item (D) we take $x_i$ (the $i$-th variable dangling from the roots) and apply the fact that $\pi$ is a homomorphism, thus all the roles mentioned in $q'$ between $\pi'(\texttt{root-of}(i))$ and $\pi'(x_i)$ are preserved, with Lemma 3.8 to infer that $\pi'(x_i) \in \mathrm{Match}^{\mathcal{I}}_{q\restriction_{\texttt{SubTree}_i}}$. This concludes the proof. □

### B.6 Proof of Lemma 3.23

*Proof.* Note that since super-spoilers are $\mathcal{ALCC}^{\cap}$-KBs, they belong to $\mathcal{DL}$ by definition. For the right-to-left direction, assume towards a contradiction that $\mathcal{K} \models_{(\mathrm{fin})} q$ holds and take any (finite) $(|q|, \mathsf{ind}(\mathcal{K}))$-lff-like model $\mathcal{I}$ of $\mathcal{K} \cup \bigcup_i \mathcal{K}^{\lightning\star}_{q_i}$ (existence guaranteed by Property 3.5). By assumption we conclude $\mathcal{I} \models q$ and hence $\mathcal{I} \models q_i$ for some $1 \leq i \leq m$. But $\mathcal{I} \models \mathcal{K}^{\lightning\star}_{q_i}$, which contradicts Lemma 3.22.

For the other direction, take any (finite) $(|q|, \mathsf{ind}(\mathcal{K}))$-lff-like countermodel $\mathcal{I}$ for $\mathcal{K}$ and $q$ (guaranteed by Property 3.5). Hence, for each $1 \leq i \leq n$ we have that $\mathcal{I} \not\models q_i$ and by Lemma 3.21 we get an $\mathsf{ind}(\mathcal{K})$-super-spoiler $\mathcal{K}^{\lightning\star}_{q_i}$ for $q_i$ such that $\mathcal{I} \models \mathcal{K}^{\lightning\star}_{q_i}$. Thus $\mathcal{I} \models \mathcal{K} \cup \bigcup_i \mathcal{K}^{\lightning\star}_{q_i}$, which concludes the proof. □

### B.7 Proof of Lemma 3.27

*Proof.* For the first statement of the lemma we consider the following cases. If $\mathcal{K}$ is not (finitely) satisfiable then it entails every query. Our procedure returns `True` in this case. If $\mathcal{K}$ is (finitely) satisfiable but does not (finitely) entail $q$, then by Lemma 3.23 there are $\mathsf{ind}(\mathcal{K})$-super-spoilers $\mathcal{K}^{\lightning\star}_{q_i}$ for $q_i$ such that $\mathcal{K} \cup \bigcup_i \mathcal{K}^{\lightning\star}_{q_i}$ is (finitely) satisfiable and hence, the fourth line of the algorithm returns `False`. Otherwise, $\mathcal{K}$ is (finitely) satisfiable and (finitely) entails $q$. Thus again, by Lemma 3.23, there are no such $\mathsf{ind}(\mathcal{K})$-super-spoilers and so the (finite) satisfiability test in the 4th line of Procedure 1 will never succeed. Hence, the 5th line will be executed, returning `True`.
The second part of the lemma follows immediately from Lemma 3.25 and Lemma 3.26 and from the fact that $\mathsf{SAT}_{\mathcal{DL}}(\mathsf{poly}(\mathcal{K}))+\mathsf{exp}(|\mathsf{ind}(\mathcal{K})|+|q|) \cdot \mathsf{SAT}_{\mathcal{DL}}(\mathsf{poly}(|\mathcal{K}|+|q|))$ is bounded by $\mathsf{exp}(|\mathcal{K}|+|q|) \cdot \mathsf{SAT}_{\mathcal{DL}}(\mathsf{poly}(|\mathcal{K}|+|q|))$. □

## Appendix C. Appendix to Section 4

### C.1 Proof of Lemma 4.2

*Proof.* $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ is an $\mathsf{N}$-rooted $\Delta^{\mathcal{I}}$-forward-forest. To see that $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ is a forward-forest we observe that its domain is prefix-closed (follows from Item 1 of Definition 4.1) and that it satisfies all forests' criteria on roles (by Item 4 of Definition 4.1). The $\mathsf{N}$-rootedness follows from Item 2 of Definition 4.1.

For the second claim, it suffices to show that $\mathsf{last}$ satisfies every item from the definition of a homomorphism. The preservation of individual names from $\mathsf{N}$ by $\mathsf{last}$ follows immediately from Item 2 of Definition 4.1. Similarly, the preservation of concepts follows from Item 3 of Definition 4.1. Lastly, to show that $\mathsf{last}$ preserves roles, take any two elements $u, v \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ satisfying $(u, v) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$. We aim to prove that $(\mathsf{last}(u), \mathsf{last}(v)) \in r^{\mathcal{I}}$ holds. There are two cases to consider: either both $u, v$ are $\mathsf{N}$-named or at least one of $u \neq v$ is not named. For the first case, note that $u, v$ are single elements. Thus, we infer that $u = \mathsf{last}(u)$ and $v = \mathsf{last}(v)$ hold and by the first part of the equation in Item 4 of Definition 4.1 we conclude $(\mathsf{last}(u), \mathsf{last}(v)) \in r^{\mathcal{I}}$. Finally, if one of $u, v$ is not named, we have that $v = u \cdot \mathrm{d}$ holds for some $\mathrm{d} \in \Delta^{\mathcal{I}}$. By applying the second part of the equation in Item 4 of Definition 4.1, we infer $(\mathsf{last}(u), \mathsf{last}(v)) \in r^{\mathcal{I}}$. This finishes the proof. □

### C.2 Proof of Property 4.4

*Proof.* We will proceed with each of the items separately.

(A) The $\mathsf{last}$ function restricted to $\mathsf{N}^{\mathcal{I}}$, is actually the identity function, so it suffices to show that it is also a homomorphism. This already follows from Lemma 4.2.

(B) For the first part, take any element $\mathrm{d} \in \mathrm{C}^{\mathcal{I}}$. Then $\mathrm{d} \in \mathrm{C}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$. Similarly, if $w \in \mathrm{C}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ then $\mathsf{last}(w) \in \mathrm{C}^{\mathcal{I}}$. Hence $\mathrm{C}^{\mathcal{I}}$ is non-empty iff $\mathrm{C}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ is. For the second part, take $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$. Then if both $\mathrm{d}, \mathrm{e}$ are $\mathsf{N}$-named, then by construction $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$. Otherwise we have that $(\mathrm{d}, \mathrm{de}) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$. For the other direction assume that $(w, v) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ holds. Then we again distinguish two cases: if both $w, v$ are $\mathsf{N}$-named then $w, v \in \Delta^{\mathcal{I}}$ and we infer $(w, v) \in r^{\mathcal{I}}$ by the manual assignment of roles between named elements. Otherwise by definition we have $(\mathsf{last}(w), \mathsf{last}(v)) \in r^{\mathcal{I}}$, yielding the desired equivalence.

(C) Depending on whether both $w$ and $v$ are $\mathsf{N}$-named or not, the result follows either from the first or from the second part of the equation in Item 4 of Definition 4.1. The equality between $\mathsf{Conc}_{\mathcal{I}}(\mathsf{last}(w))$ and $\mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(w)$ follows from Item 3 of Definition 4.1.

(D) Fix $\mathsf{C}$, $\mathsf{R}$, $w$ and $\mathrm{d} := \mathsf{last}(w)$ as in the statement of Property 4.4. We first show that $\mathsf{last}$ is a bijection between the sets $\mathsf{last}[A] := \{\mathsf{last}(v) \mid v \in A\}$ and $A$ defined below:

$$\Big\{ v \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} \mid \mathsf{C} = \mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(v) \text{ and } \mathsf{R} = \mathsf{Rol}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(w, v) \Big\},$$

Surjectivity is obvious, so we focus on injectivity only. Take any $u, v \in A$ that are mapped to the same element by $\mathsf{last}$. This implies that $u = u_0\mathrm{e}$ and $v = v_0\mathrm{e}$ for some (possibly empty) words $u_0, v_0 \in (\Delta^{\mathcal{I}})^*$ and $\mathrm{e} \in \Delta^{\mathcal{I}}$. There are three cases to consider:

- Both $u_0, v_0$ are non-empty. Then $u_0, v_0$ are equal to $w$ by the fact that $(w, v) \in r^{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}$ for some $r \in \mathsf{R}$ and the second part of the equation in Item 4 of Definition 4.1. Thus $u = v$.
- Both $u_0, v_0$ are empty. Then obviously $v = u$ holds.
- One of $u_0, v_0$ is empty and the other one is not. W.l.o.g assume $u_0 = \varepsilon$ and $v \neq \varepsilon$. Then by construction of $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$ (more precisely the first part of the equation in Item 4 of Definition 4.1) we infer that $w$ is $\mathsf{N}$-named and that $u$ is $\mathsf{N}$-named, and that both $w$ and $u$ are single-element sequences. Since $v$ has length at least two, it is not named. Thus by the second part of the equation in Item 4 of Definition 4.1, we infer $w = v_0$. But this means that $v$ is a two-element sequence composed of $\mathsf{N}$-named elements, which were excluded from the domain of $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$, *cf.* Item 1 of Definition 4.1. A contradiction. So such a case is not possible.

Thus $\mathsf{last}$ is indeed a bijection between $A$ and $\mathsf{last}[A]$.

Our next claim is that the identity function is the bijection between $\mathsf{last}[A]$ and $B$ given below. From that we conclude $|A| = |B|$ (and thus the whole proof) by transitivity.

$$B := \Big\{ \mathrm{e} \in \Delta^{\mathcal{I}} \mid \mathsf{C} = \mathsf{Conc}_{\mathcal{I}}(\mathrm{e}) \text{ and } \mathsf{R} = \mathsf{Rol}_{\mathcal{I}}(\mathsf{last}(w), \mathrm{e}) \Big\}.$$

We show that $\mathsf{last}[A] \subseteq B$ and $B \subseteq \mathsf{last}[A]$. The first inclusion follows from the way we defined roles in the unravelling, *cf.* Item 4 of Definition 4.1. We the other inclusion we distinguish the cases depending on whether on not $w$ is $\mathsf{N}$-named.

- $w$ is not $\mathsf{N}$-named.
  Take any $\mathrm{e} \in B$. Then we have $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$ for some $r \in \mathsf{R}$, and hence we have that $(w, w\mathrm{e}) \in r^{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}$ (by the second item of Item 4 of Definition 4.1). Applying Item (C) of Property 4.4 we infer that $\mathrm{e} \in A$, and thus $\mathrm{e} \in \mathsf{last}[A]$.
- If $w$ is $\mathsf{N}$-named, then $w = \mathrm{d}$.
  Again, take any $\mathrm{e} \in B$. Then we have $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$ for some $r \in \mathsf{R}$. If e is $\mathsf{N}$-named then e is also $\mathsf{N}$-named in $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$ (by Item 2 of Definition 4.1). Thus, by the first part of the equation in Item 4 of Definition 4.1, we know that $(\mathrm{d}, \mathrm{e}) = (\mathsf{last}(w), \mathsf{last}(\mathrm{e}))$ belongs to $r^{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}$. By Item (C) of Property 4.4 we infer that $\mathrm{e} \in A$, and hence $\mathrm{e} \in \mathsf{last}[A]$. Otherwise, if e is not $\mathsf{N}$-named, we apply the same reasoning as in the case of unnamed $w$. This concludes the proof.

(E) Immediate from Item 1 of Definition 4.1 and Item (C) of Property 4.4. □

### C.3 Proof of Lemma 4.8

*Proof.* Suppose that there exists an undirected path $\rho := \rho_1 \dots \rho_m$ in $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ of length less than $2n$ for which $\rho_1$ and $\rho_m$ are, respectively, a leaf and the root of the same pawn component. Note that $\rho$ contains elements from at least two components, as the root-to-leaf paths in every pawn component are of length $2n$ by design. Thus by the construction of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ (especially by our way of "linking" components in Definition 4.7) we know that one of the following cases holds for any two consecutive elements $\rho_{i-1}$ and $\rho_i$ of the path $\rho$:

- $\rho_{i-1}$ and $\rho_i$ are in the same component,
- $\rho_{i-1}$ is a leaf of some pawn component, say $\mathcal{I}^{(*,*)}_{\mathbb{1},*,h}$, and $\rho_i$ is is the root of $\mathcal{I}^{(*,*)}_{\mathbb{1},*,1-h}$, or
- $\rho_i$ is a leaf of some pawn component, say $\mathcal{I}^{(*,*)}_{\mathbb{1},*,h}$, and $\rho_{i-1}$ is is the root of $\mathcal{I}^{(*,*)}_{\mathbb{1},*,1-h}$.

In particular, the above observation implies that any two consecutive elements of $\rho$ taken from different components are of different "hue". Thus $\rho$ has the shape $\rho \coloneqq \rho'_0 \mathrm{d}_1 \dots \rho_{2k} \mathrm{d}_{2k+1}$, where the even-numbered paths $\rho'_{2i}$ are (possibly single-element) leaf-to-leaf paths traversing a single component, and the odd-numbered elements $\mathrm{d}_{2i+1}$ are roots of components. But this implies that $\rho_1$ and $\rho_m = \mathrm{d}_{2k+1}$ belong to components with different "hue", contradicting the fact that they belong to the same component. Hence such a path $\rho$ does not exist. □

### C.4 Proof of Lemma 4.10

*Proof.* Let $\mathrm{c}_1$ and $\mathrm{c}_2$ be leaves from $\mathcal{N}$-upper components, and let $\mathcal{I}^{(*,*)}_{\mathbb{1},\mathrm{e},*}$ be the component to which $\mathrm{c}_1$ belongs (the proof for the case when $\mathrm{c}_1$ is included in the king component is completely analogous). We closely follow the proof of Lemma 4.8. Among other things, we proved there that there is a path $\rho$ between $\mathrm{c}_1$ and $\mathrm{c}_2$ of length at most $n$, having the form $\rho \coloneqq \rho'_0 \mathrm{d}_1 \dots \rho_{2k} \mathrm{d}_{2k+1} \rho_{2k+2}$, where the even-numbered paths $\rho'_{2i}$ are (possibly single-element) leaf-to-leaf paths contained a single component, and the odd-numbered elements $\mathrm{d}_{2i+1}$ are roots of components. By assumption, all the elements from $\rho'_0$ are in $\mathcal{I}^{(*,*)}_{\mathbb{1},\mathrm{e},*}$. Suppose that have already shown that the elements of $\rho'_i$ belong to $\mathcal{I}^{(*,*)}_{\mathbb{1},\mathrm{e},*}$ (for parameters $*$ possibly different from the initial ones), and let us establish the same for the elements from $\rho'_{i+2}$. Since the elements from $\rho'_i$ belong to $\mathcal{I}^{(*,*)}_{\mathbb{1},\mathrm{e},*}$, the last element of $\rho'_i$ belongs to $\mathcal{I}^{(*,*)}_{\mathbb{1},\mathrm{e},*}$. By the linking process (*i.e.* Item (ii) of Definition 4.7), we conclude that $\mathrm{d}_{i+1}$ is the root of some $\mathcal{I}^{(*,\mathrm{e})}_{\mathbb{1},*,*}$. Analogously, this implies that the first element of $\rho'_{i+2}$ belongs to $\mathcal{I}^{(*,*)}_{\mathbb{1},\mathrm{e},*}$, yielding that all the elements of $\rho'_{i+2}$ are in $\mathcal{I}^{(*,*)}_{\mathbb{1},\mathrm{e},*}$. Hence, by induction, $\mathrm{c}_1$ and $\mathrm{c}_2$ belong to copies of the same component. □

### C.5 Proof of Lemma 4.11

*Proof.* As we already discussed shortly before the proof, the fact that $\mathsf{glue}(\mathcal{N})$ is a forward-tree follows from the linking process of components (Definition 4.7), and it is immediate to see that the identity function is a homomorphism from $\mathsf{glue}(\mathcal{N})$ to $\mathcal{N}$. To craft a homomorphism from $\mathcal{N}$ to $\mathsf{glue}(\mathcal{N})$, let $\mathfrak{h}$ be a mapping that serves as the identity function of $\mathcal{N}$-lower components and a function that maps elements from $\mathcal{N}$-upper components to their corresponding ones in the $\mathsf{glue}(\mathcal{N})$-upper component. Note that $\mathfrak{h}$ is well-defined as all $\mathcal{N}$-upper components are isomorphic (as provided by Lemma 4.10). Moreover, $\mathfrak{h}$ maps $\ell$-th leaves of upper components to the $\ell$-th leaf of their isomorphic copy. We would like to point out that when the only $\mathcal{N}$-component is the king component, then $\mathfrak{h}$ is the identity function, and hence a homomorphism. Thus for the rest of the proof we assume that all $\mathcal{N}$-lower components are pawn components. To see that $\mathfrak{h}$ is a homomorphism, we take any pair $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{N}}$ and show that $(\mathfrak{h}(\mathrm{d}), \mathfrak{h}(\mathrm{e})) \in r^{\mathsf{glue}(\mathcal{N})}$ holds. Note that by construction of $\mathcal{I}^{\vec{\pi}}_{\mathsf{N}}$ only the following cases can occur:

- Both $\mathrm{d}, \mathrm{e}$ are in the same component.
  Then we either conclude by the fact that $\mathfrak{h}$ is the identity function (the case of $\mathcal{N}$-lower components) or by the fact that any two $\mathcal{N}$-upper components are isomorphic.
- The element d is a leaf of some component and e is the root of some component. Hence, by invoking Item (i) of Definition 4.7, we know that there exists parameters $\ell$ and $h$ for which the element d is the $\ell$-th leaf of $\mathcal{I}^{(*,*)}_{\mathbb{1},\mathrm{c},h}$, and c is the root of $\mathcal{I}^{(\ell,\mathrm{c})}_{\mathbb{1},\mathsf{orig}_{\mathcal{I}}(\mathrm{e}),1-h}$.
  We actually know more: *every* $\ell$-th leaf $\mathrm{d}'$ of every $\mathcal{I}^{(*,*)}_{\mathbb{1},\mathrm{c},h}$ component satisfies $(\mathrm{d}', \mathrm{e}) \in r^{\mathcal{N}}$. In particular, this implies that between the $\ell$-th leaf $\mathrm{d}'$ of the unique $\mathcal{N}$-upper component of $\mathsf{glue}(\mathcal{N})$ and e there is an $r^{\mathsf{glue}(\mathcal{N})}$-role connection. Thus $(\mathfrak{h}(\mathrm{d}), \mathfrak{h}(\mathrm{e})) \in r^{\mathsf{glue}(\mathcal{N})}$ holds.

This concludes the proof. □

### C.6 Proof of Theorem 4.13

*Proof.* Take any finitely satisfiable $\mathcal{DL}$-KB $\mathcal{K}$, any of its finite models $\mathcal{I}$, and any positive integer $n \in \mathbb{N}$. It suffices to show that there exists an $(n, \mathsf{ind}(\mathcal{K}))$-lff-like model $\mathcal{J} \models \mathcal{K}$.

Take $\mathcal{J} := \mathcal{I}^{\vec{n}}_{\mathsf{ind}(\mathcal{K})}$. By preservation under scattered forward unravellings, we infer $\mathcal{J} \models \mathcal{K}$. Together with Corollary 4.12 we derive that $\mathcal{J}$ is a finite $(n, \mathsf{ind}(\mathcal{K}))$-lff interpretation that covers $\mathcal{I}$, which concludes the proof. □

### C.7 Proof of Property 4.14

*Proof.* We proceed with all items one by one, showing their satisfaction.

- Proof of Item (A) and Item (B) of Property 4.14
  By the construction of the king component in Definition 4.5, we see that $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ contains an isomorphic copy of $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$ restricted to all words of length at most two. Thus by applying Item (A) and Item (B) of Property 4.4, we are done.
- Proof of Item (C) of Property 4.14
  Take any $w, v$ and suppose that $(w, v) \in r^{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}$ holds for some $r \in \mathbf{N_R}$. By Item (C) of Property 4.4 it suffices to show that:

  $$\mathsf{Conc}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(\mathsf{orig}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(w)) = \mathsf{Conc}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(w) \text{ and } \mathsf{Rol}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(\mathsf{orig}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(w), \mathsf{orig}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(v)) = \mathsf{Rol}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(w, v).$$

  For the equality between sets of concepts, this follows from the fact that $w$ is just an isomorphic copy of $\mathsf{orig}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(w)$, according to Definition 4.6. For the equality between sets of roles, we consider two cases:

  (a) $w$ and $v$ are in the same component. Then we are again done by isomorphism between copies and selected fragments of $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$.
  (b) $w$ and $v$ are in different components, implying that $w$ is a leaf of some component, while $v$ is the root of some components. We then conclude by our linking process described in Definition 4.7.

- Proof of Item (D) of Property 4.14
  Take any $w \in \Delta^{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}$, non-empty set of role names $\mathsf{R} \subseteq \mathbf{N_R}$, and any set of concept

names $\mathsf{C} \subseteq \mathbf{N_C}$. We consider two cases. In the first one, we assume that $w$ is a leaf of a component. Then by Item (D) of Property 4.4 it suffices to establish a bijection between the sets

$$\Big\{v \in \Delta^{\mathcal{I}^{\vec{n}}_{\mathsf{N}}} \mid \mathsf{Conc}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(v) = \mathsf{C} \text{ and } \mathsf{Rol}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(w, v) = \mathsf{R}\Big\}, \text{and}$$

$$\Big\{v \in \Delta^{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}} \mid \mathsf{Conc}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(v) = \mathsf{C} \text{ and } \mathsf{Rol}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(\mathsf{orig}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(w), v) = \mathsf{R}\Big\},$$

which exists by the fact that components are isomorphic to neighbourhoods of $\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}$. From now on assume that $w$ is not a leaf of a component. It suffices to prove that

$$A := \Big\{v \in \Delta^{\mathcal{I}^{\vec{n}}_{\mathsf{N}}} \mid \mathsf{Conc}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(v) = \mathsf{C} \text{ and } \mathsf{Rol}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(w, v) = \mathsf{R}\Big\}, \text{and}$$

$$B := \Big\{\mathrm{d} \in \Delta^{\mathcal{I}} \mid \mathsf{Conc}_{\mathcal{I}}(\mathrm{d}) = \mathsf{C} \text{ and } \mathsf{Rol}_{\mathcal{I}}(\mathsf{orig}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}(w), \mathrm{d}) = \mathsf{R}\Big\}.$$

are equicardinal. Without loss of generality let us assume that $w$ is a member of a pawn component (the proof for the king component is analogous), which implies that $w$ is the $\ell$-th leaf of $\mathcal{I}^{(*,*)}_{\blacksquare,\mathrm{e},h}$ for some $\mathrm{e} \in \Delta^{\mathcal{I}}$ and $h \in \{0, 1\}$. To show equicardinality of $A$ and $B$, we employ the theorem by Cantor-Bernstein and establish that:

- $\mathsf{orig}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}$ is an injection from $A$ to $B$.
  The fact that $\mathsf{orig}_{\mathcal{I}^{\vec{\omega}}_{\mathsf{N}}}$ is a function follows by the linking process of Definition 4.7. For injectivity, it suffices to see that each $v \in A$ is a root of some pawn component $\mathcal{I}^{(\ell,\mathrm{e})}_{\blacksquare,*,1-h}$ and that per each $\mathrm{d} \in \Delta^{\mathcal{I}}$ such a component $\mathcal{I}^{(\ell,\mathrm{e})}_{\blacksquare,\mathrm{d},1-h}$ is unique.
- There is an injection from $B$ to $A$. The mapping $f$ that assigns to each $\mathrm{d} \in B$ the root of $\mathcal{I}^{(\ell,\mathrm{e})}_{\blacksquare,\mathrm{d},1-h}$ is the desired injection. The fact that $f$ is a function follows again from the linking process, while the injectivity of $f$ is due to the uniqueness of $\mathcal{I}^{(\ell,\mathrm{e})}_{\blacksquare,\mathrm{d},1-h}$.

- Proof of Item (E) of Property 4.14
  For any $w \in \Delta^{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}$ we have that $w$ and $\mathsf{orig}_{\mathcal{I}}(w)$ are directed-path-equivalent, that is:
  - If $\rho$ with $\rho_1 := w$ is a (possibly infinite) directed path in $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ then $\rho'$, defined as $\rho'_i := \mathsf{orig}_{\mathcal{I}}(\rho_i)$ for all $i$, is a directed path in $\mathcal{I}$ such that for all $i$ we have $\mathsf{Conc}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(\rho_i) = \mathsf{Conc}_{\mathcal{I}}(\rho'_i)$, and for all $i > 1$ we have $\mathsf{Rol}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(\rho_{i-1}, \rho_i) = \mathsf{Rol}_{\mathcal{I}}(\rho'_{i-1}, \rho'_i)$.
  - If $\rho$ with $\rho_1 := \mathsf{orig}_{\mathcal{I}}(w)$ is a (possibly infinite) directed path in $\mathcal{I}$ then $\rho'$ defined as:
    (i) $\rho'_1 := w$, and
    (ii) $\rho'_i := \rho_i$ if both $\rho'_{i-1}$ and $\rho_i$ are $\mathsf{N}$-named,
    (iii) $\rho'_i := \rho'_{i-1}\rho_i$ if $\rho'_{i-1}$ is not a leaf of the component,
    (iv) $\rho'_i$ is the root of $\mathcal{I}^{(\ell,\mathrm{e})}_{\blacksquare,\rho_i,1-h}$ if $\rho'_{i-1}$ is the $\ell$-th leaf of $\mathcal{I}^{(*,*)}_{\blacksquare,\mathrm{e},h}$ for some $\mathrm{e} \in \Delta^{\mathcal{I}}$ and $h \in \{0, 1\}$,
    (v) $\rho'_i$ is the root of $\mathcal{I}^{(\ell,\blacktriangle)}_{\blacksquare,\rho_i,0}$ if $\rho'_{i-1}$ is the $\ell$-th leaf of $\mathcal{I}_{\blacktriangle}$,
    is a directed path in $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ such that for all $i$ we have $\mathsf{Conc}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(\rho'_i) = \mathsf{Conc}_{\mathcal{I}}(\rho_i)$, and for all $i > 1$ we have $\mathsf{Rol}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(\rho'_{i-1}, \rho'_i) = \mathsf{Rol}_{\mathcal{I}}(\rho_{i-1}, \rho_i)$.

Once the construction of $\rho'$ from $\rho$ (and vice versa) is given, its correctness follows from Item (C) of Property 4.14.

□

### C.8 Proof of Lemma 4.16

*Proof.* Let $\delta$ be a semi-restricted cardinality constraint of the form:

$$\delta := \mathrm{N}_1 \cdot x_{\mathrm{C}_1} + \ldots + \mathrm{N}_k \cdot x_{\mathrm{C}_k} + M \leq \mathrm{N}_{k+1} \cdot x_{\mathrm{C}_{k+1}} + \ldots + \mathrm{N}_{k+\ell} \cdot x_{\mathrm{C}_{k+\ell}},$$

and let $\mathfrak{s}$ be a solution for $\delta$, *i.e.* we have that the inequality

$$(\heartsuit):\ \mathrm{N}_1 \cdot \mathfrak{s}(x_{\mathrm{C}_1}) + \ldots + \mathrm{N}_k \cdot \mathfrak{s}(x_{\mathrm{C}_k}) + M \leq \mathrm{N}_{k+1} \cdot \mathfrak{s}(x_{\mathrm{C}_{k+1}}) + \ldots + \mathrm{N}_{k+\ell} \cdot \mathfrak{s}(x_{\mathrm{C}_{k+\ell}})$$

as well as the inequality (obtained by weakening ($\heartsuit$))

$$(\clubsuit):\ \mathrm{N}_1 \cdot \mathfrak{s}(x_{\mathrm{C}_1}) + \ldots + \mathrm{N}_k \cdot \mathfrak{s}(x_{\mathrm{C}_k}) \leq \mathrm{N}_{k+1} \cdot \mathfrak{s}(x_{\mathrm{C}_{k+1}}) + \ldots + \mathrm{N}_{k+\ell} \cdot \mathfrak{s}(x_{\mathrm{C}_{k+\ell}})$$

evaluate to true.

Take a positive natural number $n > 1$, and let $\mathfrak{s}_n$ be a mapping that maps every variable $x$ to $n \cdot \mathfrak{s}(x)$. We claim that $\mathfrak{s}_n$ is also a solution for $\delta$. By multiplying both sides of ($\clubsuit$) by $(n-1)$, adding the inequality ($\heartsuit$) and simplifying the terms we get:

$$\mathrm{N}_1 \cdot n \cdot \mathfrak{s}(x_{\mathrm{C}_1}) + \ldots + \mathrm{N}_k \cdot n \cdot \mathfrak{s}(x_{\mathrm{C}_k}) + M \leq \mathrm{N}_{k+1} \cdot n \cdot \mathfrak{s}(x_{\mathrm{C}_{k+1}}) + \ldots + \mathrm{N}_{k+\ell} \cdot n \cdot \mathfrak{s}(x_{\mathrm{C}_{k+\ell}}),$$

which, by definition of $\mathfrak{s}_n$, is clearly equal to

$$\mathrm{N}_1 \cdot \mathfrak{s}_n(x_{\mathrm{C}_1}) + \ldots + \mathrm{N}_k \cdot \mathfrak{s}_n(x_{\mathrm{C}_k}) + M \leq \mathrm{N}_{k+1} \cdot \mathfrak{s}_n(x_{\mathrm{C}_{k+1}}) + \ldots + \mathrm{N}_{k+\ell} \cdot \mathfrak{s}_n(x_{\mathrm{C}_{k+\ell}}).$$

This justifies our claim.

Next, let $\mathfrak{s}$ be a solution for an ERCBox $\mathcal{E}$. Then $\mathfrak{s}_n$ (defined as above) is a solution for at least the same semi-restricted cardinality constraints from $\mathcal{E}$ as $\mathfrak{s}$. As $\mathcal{E}$ is negation-free, we conclude that $\mathfrak{s}_n$ is also a solution for $\mathcal{E}$, finishing the proof. □

### C.9 Proof of Lemma 4.20

*Proof.* Let $\mathcal{I}$ be a finite model of an ERCBox $\mathcal{E}$, and let $\mathbf{C}_\mathcal{I} := \{\mathrm{C_d} \mid \mathrm{d} \in \Delta^\mathcal{I}\}$ be a set of fresh concept names per each domain element $\mathrm{d} \in \mathcal{I}$. W.l.o.g. $\mathcal{I}$ interprets concept names from $(\mathbf{N_C} \setminus \mathbf{C}_\mathcal{I})$ as empty sets. Take $\mathcal{J}$ to be the unique interpretation whose $(\mathbf{N_C} \setminus \mathbf{C}_\mathcal{I})$-reduct is $\mathcal{I}$, and that interprets $\mathrm{C_d}$ as singletons $\{\mathrm{d}\}$. This implies that the interpretations of fresh symbols from $\mathbf{C}_\mathcal{I}$ in $\mathcal{J}$ are pairwise-disjoint, which we will exploit later. Moreover, observe that $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ is the $(\mathbf{N_C} \setminus \mathbf{C}_\mathcal{I})$-reduct of $\mathcal{J}^{\vec{n}}_{\mathsf{N}}$.

We rewrite $\mathcal{E}$ to make it aware of fresh concept symbols. We proceed with each $\delta \in \mathcal{E}$:

$$\delta := \mathrm{N}_1 \cdot x_{\mathrm{C}_1} + \ldots + \mathrm{N}_k \cdot x_{\mathrm{C}_k} + M \leq \mathrm{N}_{k+1} \cdot x_{\mathrm{C}_{k+1}} + \ldots + \mathrm{N}_{k+\ell} \cdot x_{\mathrm{C}_{k+\ell}}$$

and replace them by $\delta'$ given below:

$$\mathrm{N}_1 \cdot \left( \sum_{\mathrm{d} \in \mathrm{C}_1^\mathcal{I}} x_{\mathrm{C_d}} \right) + \ldots + \mathrm{N}_k \cdot \left( \sum_{\mathrm{d} \in \mathrm{C}_k^\mathcal{I}} x_{\mathrm{C_d}} \right) + M \leq \mathrm{N}_{k+1} \cdot \left( \sum_{\mathrm{d} \in \mathrm{C}_{k+1}^\mathcal{I}} x_{\mathrm{C_d}} \right) + \ldots + \mathrm{N}_{k+\ell} \cdot \left( \sum_{\mathrm{d} \in \mathrm{C}_{k+\ell}^\mathcal{I}} x_{\mathrm{C_d}} \right)$$

Moreover, per each concept A mentioned in $\mathcal{E}$ we append the inequalities:

$$\sum_{\mathrm{d}\in \mathrm{A}^{\mathcal{I}}} x_{\mathrm{C_d}} \leq x_{\mathrm{A}} \qquad\qquad x_{\mathrm{A}} \leq \sum_{\mathrm{d}\in \mathrm{A}^{\mathcal{I}}} x_{\mathrm{C_d}}$$

Call the resulting ERCBox $\mathcal{E}'$. Since concept names from $\mathbf{C}_{\mathcal{I}}$ are interpreted in $\mathcal{J}$ as pairwise-disjoint singleton sets whose union is $\Delta^{\mathcal{J}}$, and because $\mathcal{E}'$ is just a finer-grained description of $\mathcal{E}$, it should be clear that $\mathcal{J} \models \mathcal{E}'$ if and only if $\mathcal{I} \models \mathcal{E}$. Note that by construction, the variables $x_{\mathrm{C_d}}$ and $x_{\mathrm{C_e}}$ for different $\mathrm{d} \neq \mathrm{e}$ are linearly independent, in the sense that the duplication an element from concept $\mathrm{C_d}$ does not influence the total number of elements in $\mathrm{C_e}$ and vice versa (but can, of course, influence sizes of other concepts).

Consider a solution $\mathfrak{s} : x_{\mathrm{C}} \mapsto |\mathrm{C}^{\mathcal{J}}|$ for $\mathcal{E}'$. By Lemma 4.16, we infer that the mapping $\mathfrak{s}' : x_{\mathrm{C}} \mapsto \left(|\Delta^{\mathcal{J}^{\vec{n}}_{\mathsf{N}}}| + 1\right) \cdot \mathfrak{s}(x_{\mathrm{C}})$ is also a solution for $\mathcal{E}'$. By the fact that unravellings preserve concepts (*cf.* Item (B) of Property 4.14) we know that for each variable $x_{\mathrm{C_d}}$ mapped by $\mathfrak{s}$ to a positive value (resp. 0), the concepts $\mathrm{C_d}$ are non-empty (resp. empty) in the unravelling. Thus our proof plan is simple: we are going to duplicate elements d to make the total number in concept $\mathrm{C_d}$ in $\mathcal{J}^{\vec{n}}_{\mathsf{N}}$ equal to $\mathfrak{s}'(x_{\mathrm{C_d}})$. Formally, we take $S$ defined as:

$$S := \Big\{ \left(\mathrm{d}, \mathfrak{s}'(x_{\mathrm{C_d}}) - |\mathrm{C}_{\mathrm{d}}^{\mathcal{J}^{\vec{n}}_{\mathsf{N}}}|\right) \mid \mathrm{d} \in \Delta^{\mathcal{I}}, \left(\mathfrak{s}'(x_{\mathrm{C_d}}) - |\mathrm{C}_{\mathrm{d}}^{\mathcal{J}^{\vec{n}}_{\mathsf{N}}}|\right) > 0 \Big\}.$$

Note that value $\left(\mathfrak{s}'(x_{\mathrm{C_d}}) - |\mathrm{C}_{\mathrm{d}}^{\mathcal{J}^{\vec{n}}_{\mathsf{N}}}|\right)$ is non-negative, as $\mathfrak{s}'$ was taken to be "sufficiently large". Fact 4.19 together with routine calculations yields $(\mathcal{J}^{\vec{n}}_{\mathsf{N}})_{+S} \models \mathcal{E}'$. By the presence of additional inequalities relating concepts from $\mathbf{C}_{\mathcal{I}}$ with concepts appearing in $\mathcal{E}$, we conclude that $(\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S} \models \mathcal{E}$ holds. □

### C.10 Proof of Property 4.22

*Proof sketch..* For brevity, let $\mathcal{J} \coloneqq (\mathcal{I}^{\vec{n}}_{\mathsf{N}})_{+S}$. Note that, by construction of $S$-duplication, the interpretation $\mathcal{J}$ restricted to the elements from $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ and the interpretation $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ are isomorphic. Moreover, the process of $S$-duplication does not affect named elements of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$. Hence, by transitivity and Property 4.14, we establish Item (A) of Property 4.22. For the remaining properties we proceed as follows. Take any element $w$ and its copy $w'$. By construction (see: Definition 4.17) we know that $\mathsf{Conc}_{\mathcal{J}}(w) = \mathsf{Conc}_{\mathcal{J}}(w')$ as well as for any $v \in \Delta^{\mathcal{J}}$ we have $\mathsf{Rol}_{\mathcal{J}}(w, v) = \mathsf{Rol}_{\mathcal{J}}(w', v)$. This also implies that for any directed path $\rho$ in $\mathcal{J}$ we have that $w\rho$ is a directed path in $\mathcal{J}$ if and only if $w'\rho$ is. The above properties, by transitivity and Property 4.14 imply the satisfaction of Items (B)–(E) of Property 4.22. □

### C.11 Proof of Theorem 4.23

*Proof.* Take any finitely satisfiable $\mathcal{DL}$-KB $\mathcal{K}$, any of its finite models $\mathcal{I}$ and any positive integer $n \in \mathbb{N}$. It suffices to show that there exists an $(n, \mathsf{ind}(\mathcal{K}))$-lff-like model $\mathcal{J} \models \mathcal{K}$. Take $\mathcal{J} := \mathcal{I}^{\vec{n}}_{\mathsf{N}}$. By the assumption about preservation under scattered unravelling by rebalancing, we infer the existence of a set $S$, so that $\mathcal{J}_{+S} \models \mathcal{K}$. Thus, what remains to be done is to show that $\mathcal{J}_{+S}$ that finite $(n, \mathsf{N})$-lff interpretation that covers $\mathcal{I}$, but this follows from Lemma 4.21. □

## Appendix D. Appendix to Section 5

### D.1 Proof of Lemma 5.2

*Proof.* Let $\mathcal{U}$ be an $n$-configuration unit. We will proceed with all axioms $\alpha$ of $\mathcal{K}^n_{unit}$ and show that $\mathcal{U} \models \alpha$.

1. Note that $\Delta^{\mathcal{U}}$ is a set of all binary words of length $\leq n$ and by definition we have that $w \in \mathrm{Lvl}_i^{\mathcal{U}}$ iff $|w| = i$. Since every word $w$ has a *unique* length, it follows that $\mathcal{U} \models$ (LvlCov) and that $\mathcal{U} \models$ (LvlDisj[i,j]) for any indices $0 \leq j < i \leq n$.
2. By definitions of $\mathrm{Lvl}_n^{\mathcal{U}}$ and $next^{\mathcal{U}}$, we immediately conclude $\mathcal{U} \models$ (leaves-next-loop). Moreover, for any role name $s$ from $\mathbf{R}_{unit} \setminus \{next\}$, we conclude $\mathcal{U} \models \exists s.\mathsf{Self}$ from the fact that the set $\{(w, w) \mid\in \Delta^{\mathcal{U}}\}$ is explicitly stated as a part of $s^{\mathcal{U}}$ in its definition. Thus, we infer $\mathcal{U} \models$ (all-loops-but-next).
3. The satisfaction of (LRCov) and (LRDisj) by $\mathcal{U}$ is due to the equality $\mathrm{R}^{\mathcal{U}} = \Delta^{\mathcal{U}} \setminus \mathrm{L}^{\mathcal{U}}$.
4. Suppose $i < n$. We will show that $\mathcal{U} \models$ (LsuccLvl[i]) (the satisfaction of (RsuccLvl[i]) is analogous). Hence, take any $w \in \mathrm{Lvl}_i^{\mathcal{U}}$. Then $w$ (by definition of $\ell_{i+1}$) has exactly two $\ell_{i+1}$-successors: $w$ and $w0$. Moreover, by definition of $\mathrm{Lvl}_{i+1}^{\mathcal{U}}$ and $\mathrm{L}^{\mathcal{U}}$ we conclude that $w0 \in \mathrm{Lvl}_{i+1}^{\mathcal{U}} \cap \mathrm{L}^{\mathcal{U}}$ and $w \notin \mathrm{Lvl}_{i+1}$. Hence, $\mathcal{U} \models$ (LsuccLvl[i]) holds, since $w0$ is the required (only) witness for the $\exists$- ($\forall$-) restriction.
5. To see $\mathcal{U} \models$ (LBitZero[i]), it suffices to take any element $w \in \mathrm{Lvl}_i^{\mathcal{U}} \cap \mathrm{L}^{\mathcal{U}}$. By the first inclusion we infer that $|w| = i$ and by the second that the last letter of $w$ is 0. Hence, we are done by the definition of $(\mathrm{Ad}_i^b)^{\mathcal{U}}$. Similarly, we get $\mathcal{U} \models$ (RBitOne[i]). The property $\mathcal{U} \models$ (AdDisj[i]) is due to the fact that words from $\mathcal{U}$ carry only one letter per position. Next, to show $\mathcal{U} \models$ (AdLvlDisj[i,j]) for any $1 \leq j < i \leq n$ it suffices to see that, by definition, $\mathrm{Lvl}_j^{\mathcal{U}}$ contains words of length $= j$ and $(\mathrm{Ad}_i^b)^{\mathcal{U}}$ contains words of length $\geq i$. Thus their intersection is empty, implying the satisfaction of (AdLvlDisj[i,j]). Finally, we need to prove $\mathcal{U} \models$ (PropBit[i]) for $1 \leq i \leq n$. To this end, note that $w \in (\mathrm{Ad}_i^b)^{\mathcal{U}}$ is equivalent to saying that $|w| \geq i$ and that the $i$-th letter of $w$ is $b$. Now observe that, by definition, that for every $s \in \mathbf{R}_{unit} \setminus \{next\}$ we have that any $s$-successor of $w$ can only be $w$, $w0$, or $w1$ (if $i \neq 0$). In any case, such a successor has length $\geq i$ and has its $i$-th letter equal to $b$. Thus its membership in $(\mathrm{Ad}_i^b)^{\mathcal{U}}$ follows, finishing the proof. □

### D.2 Proof of Lemma 5.3

*Proof.* Let $\mathcal{U}$ be an $n$-configuration unit with $\varepsilon \in \mathrm{L}^{\mathcal{U}}$ iff $\mathrm{d} \in \mathrm{L}^{\mathcal{I}}$, and that interprets of all role and concept names outside $\mathbf{R}_{unit} \cup \mathbf{C}_{unit}$ as empty sets. It is obvious that exactly one such unit exists. In what follows, we are going to define a function $\mathfrak{h} : \Delta^{\mathcal{U}} \to \Delta^{\mathcal{I}}$ inductively. Denoting the restriction of $\mathfrak{h}$ to $\{0,1\}^{\leq k}$ by $\mathfrak{h}_{\leq k}$, our inductive assumption states, for a given $k \leq n$, that $\mathfrak{h}_{\leq i}$ is defined for all $i < k$ and $\mathfrak{h}_{\leq i}$ is a homomorphism from $\mathcal{U}\restriction_{\{0,1\}^{\leq k}}$ to $\mathcal{I}$, *i.e.* the restriction of $\mathcal{U}$ to the set $\{0,1\}^{\leq k}$.

We first set $\mathfrak{h}(\varepsilon) = \mathrm{d}$. It is immediate to check that $\mathfrak{h}_{\leq 0}$ is indeed a homomorphism (by $\mathcal{I} \models$ (all-loops-but-next) and our assumptions on $\varepsilon \in \mathrm{L}^{\mathcal{U}}$, and on empty interpretations of symbols outside $\mathbf{R}_{unit} \cup \mathbf{C}_{unit}$).

For the inductive step, suppose that the assumption holds for some $1 \leq k \leq n$ and take a word $w \in \{0,1\}^{k-1}$. We are going to define $\mathfrak{h}(w0)$ as follows (the case of $\mathfrak{h}(w1)$ is

symmetric). Note that since $\mathfrak{h}(w) \in \mathrm{Lvl}_{k-1}^{\mathcal{I}}$ (by the fact that $\mathfrak{h}_{\leq k-1}$ is a homomorphism) and since $\mathcal{I} \models$ (LsuccLvl[i]) (for $i$ equal to $k{-}1$) we conclude the existence of $\mathrm{d}' \in \mathrm{Lvl}_k^{\mathcal{I}}$ satisfying $(\mathfrak{h}(w), \mathrm{d}') \in \ell_k^{\mathcal{I}}$. Note that also $\mathrm{d}' \in \mathrm{L}^{\mathcal{I}} \cap (\mathrm{Ad}_k^0)^{\mathcal{I}}$ holds (by $\mathcal{I} \models$ (LsuccLvl[i]) and $\mathcal{I} \models$ (LBitZero[i]) with $i = k$). Thus, we simply let $\mathfrak{h}(w0) := \mathrm{d}'$.

What remains to be shown is the fact that $\mathfrak{h}_{\leq k}$ is a homomorphism from $\mathcal{U}\restriction_{\{0,1\}^{\leq k}}$ to $\mathcal{I}$. We already know that $\mathfrak{h}_{\leq k-1}$ preserves concepts and roles, thus we can focus on concepts and roles involving words of length $k$. Hence, take any $w$ of length $k$ and proceed with concepts first. Let A be any concept name and assume that $w \in \mathrm{A}^{\mathcal{U}}$. Our goal is to show that $\mathfrak{h}(w) \in \mathrm{A}^{\mathcal{I}}$. The cases of $\mathrm{A} \in \{\mathrm{Lvl}_k, \mathrm{L}, \mathrm{R}, \mathrm{Ad}_k^b\}$ follow immediately from the construction (see the discussion while defining them). The cases of $\mathrm{A} = \mathrm{Lvl}_j$ with $j \neq k$ and $\mathrm{A} = \mathrm{Ad}_i^b$ with $i \leq k$ cannot happen by the definition of $n$-configuration unit. Thus the only cases left are these with $\mathrm{A} = \mathrm{Ad}_i^b$ with $i < k$. But this is easy: let $w = uv$ with $|v| = 1$. By definition of $\mathcal{U}$ we have that $u \in (\mathrm{Ad}_i^b)^{\mathcal{U}}$. Since $\mathfrak{h}_{\leq k-1}$ is a homomorphism, we infer $\mathfrak{h}(u) \in (\mathrm{Ad}_i^b)^{\mathcal{I}}$ and, by $\mathcal{I} \models$ (PropBit[i]), we conclude $\mathfrak{h}(w) \in (\mathrm{Ad}_i^b)^{\mathcal{I}}$. Now we proceed with the case of role preservation by $\mathfrak{h}$. Reasoning analogously, we may focus on roles $s$ from $\mathbf{R}_{unit}$ and involving $w$ only. Thus, by definition of $\mathcal{U}$, the only cases are self-loops (that follows by $\mathcal{U} \models$ (all-loops-but-next), (leaves-next-loop)) and the roles $\ell_k^{\mathcal{U}}$ and $r_k^{\mathcal{U}}$ between the parent of $w$ and $w$, that follow from the construction.

This finishes the induction, implying that $\mathfrak{h}$ is indeed a homomorphism from $\mathcal{U}$ to $\mathcal{I}$ satisfying $\mathfrak{h}(\varepsilon) = \mathrm{d}$. We conclude by showing that $\mathfrak{h}$ is injective. Ad absurdum, suppose that there are $u \neq v \in \Delta^{\mathcal{U}}$ such that $\mathfrak{h}(u) = \mathfrak{h}(v)$. If $|u| \neq |v|$ we have that $\mathfrak{h}(u) \in \mathrm{Lvl}_{|u|}^{\mathcal{I}} \cap \mathrm{Lvl}_{|v|}^{\mathcal{I}}$ (by the definition of $\mathcal{U}$ and by preservation of concepts by $\mathfrak{h}$). This contradicts $\mathcal{I} \models$ (LvlDisj[i,j]). Otherwise, $|u| = |v|$ but their $i$-th letters differ. Again, since $\mathfrak{h}$ is a homomorphism, we conclude $\mathfrak{h}(v) \in (\mathrm{Ad}_i^0)^{\mathcal{I}} \cap (\mathrm{Ad}_i^1)^{\mathcal{I}}$, which violates $\mathcal{I} \models$ (AdDisj[i]). Hence, $\mathfrak{h}$ is injective. □

### D.3 Proof of Lemma 5.4

*Proof.* For simplicity we use $s_i^{\mathcal{U}}$ as an abbreviation of $\ell_1{}^{\mathcal{U}} \circ r_1{}^{\mathcal{U}} \circ \ldots \circ \ell_i{}^{\mathcal{U}} \circ r_i{}^{\mathcal{U}}$. The proof is by induction, where the assumption is that for all $1 \leq i \leq n$ we have that all words $w$ of length at most $i$ satisfy $(\varepsilon, w) \in s_i^{\mathcal{U}}$. The base case (for $w \in \{\varepsilon, 0, 1\}$) is immediate to verify. Now take any word $w$ of length at most $i{+}1$ and consider the following two cases:

1. $|w| \leq i$. Hence, by the inductive assumption we have $(\varepsilon, w) \in s_i^{\mathcal{U}}$. Since $(w, w) \in \ell_{i+1}^{\mathcal{U}}$ and $(w, w) \in r_{i+1}^{\mathcal{U}}$, by the definition of composition we conclude $(\varepsilon, w) \in s_{i+1}^{\mathcal{U}}$.
2. $|w| = i{+}1$. Hence, $w = u0$ or $u1$ for some $|u| = i$. We focus on the first case, the second one is symmetric. By the inductive assumption we infer that $(\varepsilon, u) \in s_i^{\mathcal{U}}$. Since $(u, u0) \in \ell_{i+1}^{\mathcal{U}}$ and $(w, w) \in r_{i+1}^{\mathcal{U}}$ we conclude $(\varepsilon, w) \in s_{i+1}^{\mathcal{U}}$, which finishes the proof. □

### D.4 Proof of Lemma 5.7

*Proof.* Let $\mathcal{C}$ be a configuration tree. We proceed with all axioms $\alpha$ of $\mathcal{K}_{conf}$ showing $\mathcal{C} \models \alpha$.

1. Since $\mathcal{C}$ is an $(\mathtt{N}{+}1)$-configuration unit by definition, by Lemma 5.2 we infer $\mathcal{C} \models \mathcal{K}_{unit}^{\mathtt{N}+1}$.
2. The satisfactions $\mathcal{C} \models$ (StCov) and $\mathcal{C} \models$ (StDisj[$\mathtt{s}$,$\mathtt{s}'$]) follow immediately from the first item of Definition 5.6.

3. We have $\mathcal{C} \models$ (LetDisj) and $\mathcal{C} \models$ (LetCov) by the second item of Definition 5.6. Next, we have $\mathcal{C} \models$ (LetConDisj) by the fact that words in $\mathrm{Let}_{\mathtt{0}}^{\mathcal{C}}$ and $\mathrm{Let}_{\mathtt{1}}^{\mathcal{C}}$ have different last letters. The satisfaction of (LetConCov) by $\mathcal{C}$ is due to the 3rd property in the 3rd item of Definition 5.6. What remains to show is the satisfaction of (EncLetZero) (the proof of (EncLetOne) is symmetric), but this is due to the fact that the only $\ell_{\mathtt{N}+1}$- (resp. $r_{\mathtt{N}+1}$-) successor of any $w \in \mathrm{Let}_{\mathtt{0}}^{\mathcal{C}}$ (thus also from $\mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ by the previous statement) is $w0$ (resp. $w1$) that belong to $\mathbb{0}^{\mathcal{C}}$ (resp. $\mathbb{1}^{\mathcal{C}}$) by definition.

4. Before we proceed with the next part of the axiomatisation, we establish a few properties about concept memberships of $w_{head}$. First, we have $w_{head} \in \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ by definition (4th item). Moreover, for every $i$ we have that $w_{head}$ belongs to exactly one of $(\mathrm{Ad}_i^0)^{\mathcal{C}}, (\mathrm{Ad}_i^1)^{\mathcal{C}}$ by the definition of a unit. Hence, the choice of $w_{head}$ fixes interpretations of $\mathrm{Ad}_i^0, \mathrm{Ad}_i^1$ and, as we will see, also $\mathrm{HdPos}_i^0$ and $\mathrm{HdPos}_i^1$. Indeed, by the 5th item of Definition 5.6, whenever $w_{head} \in (\mathrm{Ad}_i^b)^{\mathcal{C}}$ then $(\mathrm{HdPos}_i^b)^{\mathcal{C}} = \mathrm{Lvl}_0^{\mathcal{C}} \cup \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ and $(\mathrm{HdPos}_i^{1-b})^{\mathcal{C}} = \emptyset$, thus also $(\mathrm{HdPos}_i^b)^{\mathcal{C}} \cap (\mathrm{HdPos}_i^{1-b})^{\mathcal{C}} = \emptyset$ and $(\mathrm{HdPos}_i^b)^{\mathcal{C}} \cup (\mathrm{HdPos}_i^{1-b})^{\mathcal{C}} = \mathrm{Lvl}_0^{\mathcal{C}} \cup \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ hold. This establishes $\mathcal{C} \models$ (HdPosDisj[i]) and $\mathcal{C} \models$ (HdPosCov[i]) for all $1 \leq i \leq \mathtt{N}$.

5. If $\varepsilon \in (\mathrm{Lvl}_0 \sqcap \mathrm{HdPos}_i^b)^{\mathcal{C}}$, then by definition $w_{head} \in (\mathrm{Ad}_i^b)^{\mathcal{C}}$. This implies $(\mathrm{HdPos}_i^b)^{\mathcal{C}} = \mathrm{Lvl}_0^{\mathcal{C}} \cup \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ and thus $\mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}} \subseteq (\mathrm{HdPos}_i^b)^{\mathcal{C}}$, which is even stronger than the meaning of the GCI (PropHdPos[i,b]). Hence, we have $\mathcal{C} \models$ (PropHdPos[i,b]) for all $1 \leq i \leq \mathtt{N}$ and $0 \leq b \leq 1$.

6. We next focus on proving $\mathcal{C} \models$ (HdHereEqualAdr) (the proof of (NoHdHereDiffrAdr) is symmetric) and $\mathcal{C} \models$ (HdHereCov). The second GCI follows by definition, hence we focus on the first one. Ad absurdum, assume that there is $w \in (\mathrm{Lvl}_{\mathtt{N}} \sqcap \bigsqcap_{i=1}^{\mathtt{N}} \bigsqcup_{b\in\{0,1\}} (\mathrm{Ad}_i^b \sqcap \mathrm{HdPos}_i^b))^{\mathcal{C}}$ but $w \notin (\mathrm{HdHere})^{\mathcal{C}}$. Since $w \in (\mathrm{Lvl}_{\mathtt{N}})^{\mathcal{C}}$ we infer $|w| = \mathtt{N}$ and, by $w \notin (\mathrm{HdHere})^{\mathcal{C}}$ and the 4th item of Definition 5.6, we infer $w \neq w_{head}$. Thus there is a position $1 \leq k \leq \mathtt{N}$ such the $k$-th letter of $w$ differs from the $k$-th letter of $w_{head}$ (called it $b$). So we have $w \notin (\mathrm{Ad}_k^b)^{\mathcal{C}}$ and $w \in (\mathrm{Ad}_k^{1-b})^{\mathcal{C}}$ (by definition of $\mathrm{Ad}_i^b$ in units). At the same time the 5th item of Definition 5.6 informs us that $(\mathrm{HdPos}_k^b)^{\mathcal{C}} = \mathrm{Lvl}_0^{\mathcal{C}} \cup \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ and $(\mathrm{HdPos}_k^{1-b})^{\mathcal{C}} = \emptyset$, which implies $w \in (\mathrm{HdPos}_k^b)^{\mathcal{C}}$ and $w \notin (\mathrm{HdPos}_k^{1-b})^{\mathcal{C}}$. This contradicts the fact that $w \in (\bigsqcup_{b\in\{0,1\}} (\mathrm{Ad}_i^b \sqcap \mathrm{HdPos}_i^b))^{\mathcal{C}}$.

7. We proceed with the last three GCIs. Satisfaction of (HdLetCov) by $\mathcal{C}$ follows by definition. For (RetrHdLet[$\mathtt{a}$]), assume that its antecedent is non-empty (which means that it is equal to $\{\varepsilon\}$). This implies, by the 4th and the last item of Definition 5.6, that $w_{head} \in \mathrm{Let}_{\mathtt{a}}^{\mathcal{C}}$. Thus, by definition, $\mathrm{HdLet}_{\mathtt{a}}^{\mathcal{C}}$ is equal to $\{\varepsilon\}$, which obviously contains $\varepsilon$. Hence, $\mathcal{C} \models$ (RetrHdLet[$\mathtt{a}$]). Finally, $\mathcal{C} \models$ (HdLetUnique[$\mathtt{a}$]) is shown as follows. If the antecedent of (HdLetUnique[$\mathtt{a}$]) is non-empty then it is equal to $\{\varepsilon\}$. Thus by the list item of Definition 5.6, we have $\mathrm{Let}_{\mathtt{a}}^{\mathcal{C}} = \{w_{head}\}$. Since $\mathrm{HdHere}^{\mathcal{C}} = \{w_{head}\}$ (by the 4th item of Definition 5.6), we conclude $\mathrm{HdHere}^{\mathcal{C}} \subseteq \mathrm{Let}_{\mathtt{a}}^{\mathcal{C}}$. Therefore $\mathcal{C} \models$ (HdLetUnique[$\mathtt{a}$]) holds, finishing the proof. □

### D.5 Proof of Lemma 5.8

*Proof.* By Lemma 5.3 there is an ($\mathtt{N}$+1)-configuration unit $\mathcal{U}$ and a homomorphism $\mathfrak{h} : \mathcal{U} \to \mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$. Moreover, as the symbols outside $\mathbf{R}_{unit} \cup \mathbf{C}_{unit}$ do not appear in Definition 5.1 we can assume that $\mathcal{U}$ interprets them as empty sets. Let $\mathcal{C} = (\Delta^{\mathcal{U}}, \cdot^{\mathcal{C}})$ be an interpretation

that is obtained from changing the meaning of concepts from $\mathbf{C}_{conf}$ as follows: for any $\mathrm{C} \in \mathbf{C}_{conf}$ we let $\mathrm{C}^{\mathcal{C}} := \{w \mid \mathfrak{h}(w) \in \mathrm{C}^{\mathcal{U}}\}$. All other symbols are interpreted as in $\mathcal{U}$. Clearly $\mathfrak{h}$ is a homomorphism from $\mathcal{C}$ into $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$. It suffices to prove that $\mathcal{C}$ is a configuration tree. This is done by routine investigation of items from Definition 5.6 and the presented GCIs.

- Let $\mathtt{s}$ be the unique state satisfying $\mathfrak{h}(\varepsilon) \in \mathrm{St}_{\mathtt{s}}^{\mathcal{I}}$: it exists by $\mathcal{I} \models$ (StCov) and is unique by $\mathcal{I} \models$ (StDisj[$\mathtt{s}$,$\mathtt{s}'$]). Hence, $(\mathrm{St}_{\mathtt{s}})^{\mathcal{C}} = \{\varepsilon\}$ holds. Moreover, $(\mathrm{St}_{\mathtt{s}'})^{\mathcal{C}} = \emptyset$ for all $\mathtt{s}' \neq \mathtt{Q}$. Note that $\varepsilon \notin (\mathrm{St}_{\mathtt{s}'})^{\mathcal{C}}$ by (StDisj[$\mathtt{s}$,$\mathtt{s}'$]) and for other elements (so from $\mathrm{Lvl}_i^{\mathcal{C}}$ for some $i > 0$) their membership in $(\mathrm{St}_{\mathtt{s}'})^{\mathcal{C}}$ would violate $\mathcal{I} \models \bigsqcup_{\mathtt{s} \in \mathtt{Q}} \mathrm{St}_{\mathtt{s}}$.
- Similarly, the equalities $(\mathrm{Lvl}_{\mathtt{N}+1})^{\mathcal{C}} = \mathbb{0}^{\mathcal{C}} \cup \mathbb{1}^{\mathcal{C}}$ and $\mathbb{0}^{\mathcal{C}} \cap \mathbb{1}^{\mathcal{C}} = \emptyset$ follow by $\mathcal{I} \models$ (LetDisj) and $\mathcal{I} \models$ (LetCov).
- We have $(\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}} \cup (\mathrm{Let}_{\mathtt{1}})^{\mathcal{C}} = \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ by $\mathcal{I} \models$ (LetConDisj) and $\mathcal{I} \models$ (LetConCov). Note that this implies $(\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}} \subseteq \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$. We next show the equality $(\heartsuit) : (\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}} = \{w \in \Delta^{\mathcal{U}} \mid w0 \in \mathbb{0}^{\mathcal{C}}, w1 \in \mathbb{1}^{\mathcal{C}}\}$ (the related equality for $\mathrm{Let}_{\mathtt{1}}$ is symmetric). Take any $w \in (\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}}$ (thus also $\in \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$). By the fact that $\mathcal{C}$ is a unit, we infer that $w0 \in \mathrm{Lvl}_{\mathtt{N}+1}^{\mathcal{C}}$ and $w1 \in \mathrm{Lvl}_{\mathtt{N}+1}^{\mathcal{C}}$ exist, and moreover $w$ is linked to them, respectively, by the roles $\ell_{\mathtt{N}+1}^{\mathcal{C}}$ and $r_{\mathtt{N}+1}^{\mathcal{C}}$. Hence, by the homomorphic assignment of concepts from $\mathbf{C}_{conf}$ and the satisfaction $\mathcal{I} \models$ (EncLetZero) we have that $w0 \in \mathbb{0}^{\mathcal{C}}$ and $w1 \in \mathbb{1}^{\mathcal{C}}$. Hence, the $\subseteq$-relationship of $(\heartsuit)$ follows. For the $\supseteq$-relationship take any $w \in \Delta^{\mathcal{U}}$ s.t. $w0 \in \mathbb{0}^{\mathcal{C}}, w1 \in \mathbb{1}^{\mathcal{C}}$ and note that $w \in \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ and $wb \in \mathrm{Lvl}_{\mathtt{N}+1}^{\mathcal{C}}$ hold. Otherwise, by the fact that $\mathcal{C}$ is an ($\mathtt{N}$+1)-configuration unit, the element $w0$ does not exist or it belongs to $\mathrm{Lvl}_i^{\mathcal{C}}$ for $i \neq \mathtt{N}+1$, which violates $\mathcal{I} \models$ (LetCov). By $\mathcal{I} \models$ (LetConCov) we know that $w \in (\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}} \cup (\mathrm{Let}_{\mathtt{1}})^{\mathcal{C}}$. If $w \in (\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}}$ holds then we are done. Thus, assume towards a contradiction that $w \in (\mathrm{Let}_{\mathtt{1}})^{\mathcal{C}}$. But then the first conjunct of the consequent of (EncLetOne) is violated, contradicting its satisfaction by $\mathcal{I}$. Hence $(\heartsuit)$ holds.
- Let $w_{head}$ be the unique $\mathtt{N}$-digit binary word whose $i$-th letter is equal to $b$ iff $\varepsilon \in (\mathrm{HdPos}_i^b)^{\mathcal{C}}$ holds. This is well defined due to $\mathcal{I} \models$ (HdPosDisj[i]) and $\mathcal{I} \models$ (HdPosCov[i]) for all $1 \leq i \leq \mathtt{N}$. It remains to show that this $w_{head}$ indeed plays the role of $w_{head}$ in the sense of Definition 5.6. Take any $1 \leq i \leq \mathtt{N}$ and let $b$ be the $i$-th letter of $w_{head}$. We will show that $(\mathrm{HdPos}_i^{1-b})^{\mathcal{C}} = \emptyset$ holds. Ad absurdum, assume that it is non-empty and contains $w$. By $\mathcal{I} \models$ (HdPosCov[i]) we have that either $w \in \mathrm{Lvl}_0^{\mathcal{C}}$ or $w \in \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$. The first case is not possible due to $\mathcal{I} \models$ (HdPosDisj[i]) (we already have that $\varepsilon \in (\mathrm{HdPos}_i^b)^{\mathcal{C}}$). Thus $w \in \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$. But then it violates $\mathcal{I} \models$ (PropHdPos[i,b]), which by Lemma 5.4 enforces that $w \in (\mathrm{HdPos}_i^b)^{\mathcal{C}}$. Hence, ($\clubsuit$:) $(\mathrm{HdPos}_i^{1-b})^{\mathcal{C}} = \emptyset$, which by (HdPosCov[i]) implies ($\clubsuit'$:) $\mathrm{Lvl}_0^{\mathcal{C}} \cup \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}} = (\mathrm{HdPos}_i^b)^{\mathcal{C}}$. Thus it follows, by definition of $w_{head}$, that $w_{head} \in (\mathrm{Ad}_i^b)^{\mathcal{C}}$ iff $w_{head} \in (\mathrm{HdPos}_i^b)^{\mathcal{C}}$, concluding that $\mathcal{C}$ satisfies the 5th item of Definition 5.6. Next, by exploiting ($\clubsuit$) and ($\clubsuit'$) and applying the satisfaction of (HdHereEqualAdr), (NoHdHereDiffrAdr) and (HdHereCov) by $\mathcal{C}$, we conclude the satisfaction of the 4th item of Definition 5.6. This, among the other properties, results in $\mathrm{HdHere}^{\mathcal{C}} = \{w_{head}\}$. Finally, let $\mathtt{a}$ be the unique letter satisfying $w_{head} \in \mathrm{Let}_{\mathtt{a}}^{\mathcal{C}}$ (it exists and is unique by $\mathcal{I} \models$ (LetConDisj), (LetConCov)). We claim that $\mathrm{HdLet}_{\mathtt{a}}^{\mathcal{C}} = \{\varepsilon\}$ and $\mathrm{HdLet}_{\mathtt{1}-\mathtt{a}}^{\mathcal{C}} = \emptyset$. Note that both $\mathrm{HdLet}_{\mathtt{a}}^{\mathcal{C}}$ and $\mathrm{HdLet}_{\mathtt{1}-\mathtt{a}}^{\mathcal{C}}$ are subsets of $\{\varepsilon\}$ by $\mathcal{I} \models$ (HdLetCov), thus it suffices to show that $\varepsilon \in \mathrm{HdLet}_{\mathtt{a}}^{\mathcal{C}}$ and $\varepsilon \notin \mathrm{HdLet}_{\mathtt{1}-\mathtt{a}}^{\mathcal{C}}$. The first property holds due to $\mathcal{I} \models$ (RetrHdLet[$\mathtt{a}$]). For the second property, towards a

contradiction assume that $\varepsilon \in \mathrm{HdLet}^{\mathcal{C}}_{\mathtt{1-a}}$. Hence, by $\mathcal{I} \models$ (HdLetUnique[a]) we conclude that $\mathrm{HdHere}^{\mathcal{C}} = \{w_{head}\} \subseteq \mathrm{Let}^{\mathcal{C}}_{\mathtt{1-a}}$. But $\mathrm{Let}^{\mathcal{C}}_{\mathtt{1-a}}$ is empty by $\mathcal{I} \models$ (LetConDisj) and the definition of a. A contradiction.

Hence the interpretation $\mathcal{C}$ is indeed a configuration tree, concluding the proof. □

### D.6 Proof of Lemma 5.10

*Proof.* Since $\mathcal{E}$ is a configuration tree by definition, by Lemma 5.7 we infer $\mathcal{E} \models \mathcal{K}_{conf}$. Hence, we may focus on the GCIs presented in this section only. For the GCIs from the 2nd group, we essentially use the same proof that we used for their "non-previous" counterparts the proof of Lemma 5.7 and thus we do not repeat it here. Satisfaction of (PHdAbvCov) and (PHdAbvDisj) follows by the 4th item of Definition 5.9. Next, to prove that also (PropPHdAbv) is satisfied by $\mathcal{E}$ (the proof for (PropNoPHdAbv) is analogous) we take any $w \in \mathrm{PHdHere}^{\mathcal{E}}$, which by definition is equal to $w_{phd}$ and see that the antecedent of the implication on the right hand of (PropPHdAbv) is satisfied only by $w_{phd}0$ and $w_{phd}1$, which are in $\mathrm{PHdAbv}^{\mathcal{E}}$ by definition. Next, to show satisfaction of (TransiCons) we assume $\varepsilon \in \mathrm{PTrns}^{\mathcal{E}}_{(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',d)}$. Then we have $\varepsilon \in \mathrm{PHdLet}^{\mathcal{E}}_{\mathtt{b}}$ (by the second to last items of Definition 5.9), $\varepsilon \in \mathrm{St}^{\mathcal{E}}_{\mathtt{s}'}$ (by the 2nd item of Definition 5.9) and that $\varepsilon \in$ "$\mathrm{PHdPos} + d = \mathrm{HdPos}$"$^{\mathcal{E}}$ (by correctness of incrementation/decrementation of binary encodings and by the 1st subitem of the 3rd item of Definition 5.9). Thus $\mathcal{E} \models$ (TransiCons). Finally, $\mathcal{E} \models$ (IC) follows directly from the last item of Definition 5.9. □

### D.7 Proof of Lemma 5.11

*Proof.* We follow the proof scheme of Lemma 5.8. By Lemma 5.8, there is a homomorphism $\mathfrak{h}$ from a configuration tree $\mathcal{C}$ to $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$. Moreover, as the symbols outside $\mathbf{R}_{unit} \cup \mathbf{C}_{unit} \cup \mathbf{C}_{conf}$ do not appear in Definition 5.6 we can assume that $\mathcal{C}$ interprets them as empty sets. Let $\mathcal{E} = (\Delta^{\mathcal{C}}, \cdot^{\mathcal{U}})$ be an interpretation that is obtained from changing the meaning of concepts from $\mathbf{C}_{enr}$ as follows: for any $\mathrm{C} \in \mathbf{C}_{enr}$ we let $\mathrm{C}^{\mathcal{E}} := \{w \mid \mathfrak{h}(w) \in \mathrm{C}^{\mathcal{C}}\}$. All other symbols are interpreted as in $\mathcal{C}$. Clearly $\mathfrak{h}$ is a homomorphism from $\mathcal{E}$ into $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$ and it suffices to show that $\mathcal{E}$ is an enriched configuration tree (we already know that it is a configuration tree), which is done by routine investigation of Definition 5.9 and the presented GCIs.

The existence of the unique concept $\mathrm{C} \in \mathbf{C}_{ptr}$ claimed in the 1st item of Definition 5.9 is provided by the first three GCIs, namely the existence is due to (TrCov) and uniqueness due to (TrInitDisj[t]) and (TrDisj[t, t′]). The 2nd item of Definition 5.9 is due to $\mathcal{I} \models$ (TransiCons) (more precisely, the first conjunct of the rhs). Similarly to the proof of Lemma 5.8, we establish the existence of a unique $w_{phd}$ and desired properties of concepts $\mathrm{PHdHere}^{\mathcal{E}}$, $\mathrm{NoPHdHere}^{\mathcal{E}}$, $(\mathrm{PHdPos}^b_i)^{\mathcal{E}}$ and $\mathrm{PHdLet}^{\mathcal{E}}_{\mathtt{a}}$. Since such a proof is nearly identical (modulo adding the "P" letter in front of some concept names) to the one from the previous section, we do not repeat the details here. Then, the fact that $w_{phd}$ satisfies $w_{phd} = w_{head} + d$ or is equal to 0 in case $\mathrm{Ini}^{\mathcal{E}} = \{\varepsilon\}$ is by, respectively, membership of $\varepsilon$ in "$\mathrm{PHdPos} + d = \mathrm{HdPos}$"$^{\mathcal{E}}$ and $(\forall \ell_1 \forall r_1 \ldots \forall \ell_{\mathtt{N}} \forall r_{\mathtt{N}}\, (\mathrm{Lvl}_{\mathtt{N}} \to \mathrm{PHdPos}^b_i))^{\mathcal{E}}$, guaranteed by $\mathcal{I} \models$ (TransiCons) and $\mathcal{I} \models$ (IC). Finally, the satisfaction of the last item of Definition 5.9 is immediate by $\mathcal{I} \models$ (IC). □

### D.8 Proof of Lemma 5.13

*Proof.* To see $\mathcal{Q} \models \mathcal{K}_{enr}$ it suffices to observe that (1) by the 2nd item of Definition 5.12 all the substructures of $\mathcal{Q}$ induced by $\{\mathfrak{w}\} \times \{0,1\}^{\leq \mathtt{N}+1}$ are isomorphic to some computation tree and hence, by Lemma 5.10 they satisfy $\mathcal{K}_{enr}$, (2) the use of roles from $\mathbf{R}_{unit} \setminus \{next\}$ is restricted to enriched configuration trees and hence $\mathcal{Q}$ satisfies all the GCIs not involving *next* and (3) the only GCI involving *next* from $\mathcal{K}_{enr}$ is (leaves-next-loop) and it is satisfied in $\mathcal{Q}$ due to the mentioned isomorphism property. Next, the satisfaction of (AConfSucc[$\mathtt{s}_a$]), (TransiUnivStateL), and (TransiUnivStateR) by $\mathcal{Q}$ is due to the 7th item (1st subitem) of Definition 5.12. Similarly, we infer that $\mathcal{Q} \models$ (EConfSucc[$\mathtt{s}_e$]) and $\mathcal{Q} \models$ (TransiExistState) by the 7th item (2nd subitem) of Definition 5.12. By the last item of Definition 5.12 we immediately conclude $\mathcal{Q} \models$ (FinConfSucc[$\mathtt{s}_f$]) and $\mathcal{Q} \models$ (NoRejectState). Hence, it remains to prove satisfaction of (TransHeadPos[$i, b$]), which is immediate by the second to last item of Definition 5.12. □

### D.9 Proof of Lemma 5.17

*Proof.* It suffices to take $q_{main} := q_{rl}[x_r, x] \wedge next(x_r, y_r) \wedge q_{rl}[y_r, y]$. Let $\mathcal{Q} \models_\pi q_{main}$. That $M_{q_{main}}$ is a superset of the set above follows from the fact that quasi-computation trees are computation units and hence, containment follows by Corollary 5.5. We now focus on the other direction. Note that by the 5th item of Definition 5.12 we know that $\pi(x_r)$ and $\pi(y_r)$ must be two distinct roots of enriched configuration trees $\mathcal{E}_{x_r}, \mathcal{E}_{y_r}$. By the 4th item of Definition 5.12 we know that the interpretation of the $r$s and $\ell$s is restricted to pairs of domain elements located inside the same enriched configuration tree (and by their definition to configuration trees and by their definition to configuration units). Since $q_{rl}$ only employs the roles $\ell_i, r_i$ and the concepts $\mathrm{Lvl}_0, \mathrm{Lvl}_{\mathtt{N}+1}$ we conclude that $q_{rl}$ has exactly the same set of matches in $\mathcal{E}_{x_r}$ as in its underlying unit. Hence, by Corollary 5.5 we know that $x$ (resp. $y$) is indeed mapped to a leaf of $\mathcal{E}_{x_r}$ (resp. to a leaf of $\mathcal{E}_{y_r}$), which finishes the proof. □

## References

Andréka, H., Németi, I., & van Benthem, J. (1998). Modal Languages and Bounded Fragments of Predicate Logic. *Journal of Philosophical Logic*, *27*(3), 217–274.

Baader, F. (2017). A New Description Logic with Set Constraints and Cardinality Constraints on Role Successors. In Dixon, C., & Finger, M. (Eds.), *Frontiers of Combining Systems - 11th International Symposium, FroCoS 2017, Brasília, Brazil, September 27-29, 2017, Proceedings*, Vol. 10483 of *Lecture Notes in Computer Science*, pp. 43–59. Springer.

Baader, F., Bednarczyk, B., & Rudolph, S. (2020). Satisfiability and Query Answering in Description Logics with Global and Local Cardinality Constraints. In De Giacomo, G., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., & Lang, J. (Eds.), *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020*, Vol. 325, pp. 616–623. IOS Press.

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

Baader, F., Horrocks, I., Lutz, C., & Sattler, U. (2017). *An Introduction to Description Logic*. Cambridge University Press.

Bárány, V., Gottlob, G., & Otto, M. (2014). Querying the Guarded Fragment. *Logical Methods in Computer Science*, *10*(2).

Bárány, V., ten Cate, B., & Segoufin, L. (2015). Guarded Negation. *Journal of the ACM*, *62*(3), 22:1–22:26.

Bednarczyk, B. (2021a). Exploiting Forwardness: Satisfiability and Query-Entailment in Forward Guarded Fragment. In Faber, W., Friedrich, G., Gebser, M., & Morak, M. (Eds.), *Logics in Artificial Intelligence - 17th European Conference, JELIA 2021, Virtual Event, May 17-20, 2021, Proceedings*, Vol. 12678 of *Lecture Notes in Computer Science*, pp. 179–193. Springer.

Bednarczyk, B. (2021b). Lutz's Spoiler Technique Revisited: A Unified Approach to Worst-Case Optimal Entailment of Unions of Conjunctive Queries in Locally-Forward Description Logics. *CoRR*, *abs/2108.05680*.

Bednarczyk, B., & Kieroński, E. (2022). Finite Entailment of Local Queries in the Z Family of Description Logics. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 5487–5494. AAAI Press.

Bednarczyk, B., & Rudolph, S. (2022). The Price of Selfishness: Conjunctive Query Entailment for ALCSelf is 2EXPTIME-Hard. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 5495–5502. AAAI Press.

Bonatti, P. A., Lutz, C., Murano, A., & Vardi, M. Y. (2008). The Complexity of Enriched Mu-Calculi. *Logical Methods in Computer Science*, *4*(3).

Bourhis, P., Krötzsch, M., & Rudolph, S. (2014). How to Best Nest Regular Path Queries. In *Informal Proceedings of the 27th International Workshop on Description Logics, Vienna, Austria, July 17-20, 2014*, Vol. 1193 of *CEUR Workshop Proceedings*, pp. 404–415. CEUR-WS.org.

Calvanese, D., Eiter, T., & Ortiz, M. (2009). Regular Path Queries in Expressive Description Logics with Nominals. In Boutilier, C. (Ed.), *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pp. 714–720.

Calvanese, D., Eiter, T., & Ortiz, M. (2014). Answering Regular Path Queries in Expressive Description Logics via Alternating Tree-Automata. *Information and Computation*, *237*, 12–55.

Chandra, A. K., Kozen, D. C., & Stockmeyer, L. J. (1981). Alternation. *Journal of the ACM*, *28*(1), 114–133.

De Giacomo, G., & Lenzerini, M. (1996). TBox and ABox Reasoning in Expressive Description Logics. In Carlucci Aiello, L., Doyle, J., & Shapiro, S. C. (Eds.), *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96), Cambridge, Massachusetts, USA, November 5-8, 1996*, pp. 316–327. Morgan Kaufmann.

De Giacomo, G., & Lenzerini, M. (1997). A Uniform Framework for Concept Definitions in Description Logics. *Journal of Artificial Intelligence Research*, *6*, 87–110.

Demri, S., & Lugiez, D. (2010). Complexity of Modal Logics with Presburger Constraints. *Journal of Applied Logics*, *8*(3), 233–252.

Eiter, T., Lutz, C., Ortiz, M., & Simkus, M. (2009). Query Answering in Description Logics with Transitive Roles. In Boutilier, C. (Ed.), *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pp. 759–764.

Eiter, T., Ortiz, M., & Simkus, M. (2012). Conjunctive query answering in the description logic SH using knots. *Journal of Computer and System Sciences*, *78*(1), 47–85.

Emerson, E. A., & Halpern, J. Y. (1985). Decision Procedures and Expressiveness in the Temporal Logic of Branching Time. *Journal of Computer and System Sciences*, *30*(1), 1–24.

Glimm, B., Lutz, C., Horrocks, I., & Sattler, U. (2008). Conjunctive Query Answering for the Description Logic SHIQ. *Journal of Artificial Intelligence Research*, *31*, 157–204.

Grädel, E. (1999). On The Restraining Power of Guards. *Journal of Symbol Logic*, *64*(4), 1719–1742.

Gutiérrez-Basulto, V., Ibáñez-García, Y. A., & Jung, J. C. (2018). Answering Regular Path Queries over SQ Ontologies. In McIlraith, S. A., & Weinberger, K. Q. (Eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 1845–1852. AAAI Press.

Horrocks, I., Kutz, O., & Sattler, U. (2006). The Even More Irresistible SROIQ. In Doherty, P., Mylopoulos, J., & Welty, C. A. (Eds.), *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, pp. 57–67. AAAI Press.

Horrocks, I., & Tessaris, S. (2000). Answering Conjunctive Queries over DL ABoxes: A Preliminary Report. In Baader, F., & Sattler, U. (Eds.), *Proceedings of the 2000 International Workshop on Description Logics (DL2000), Aachen, Germany, August 17-19, 2000*, Vol. 33 of *CEUR Workshop Proceedings*, pp. 173–182. CEUR-WS.org.

Krötzsch, M., Rudolph, S., & Hitzler, P. (2008). ELP: Tractable Rules for OWL 2. In Sheth, A. P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T. W., & Thirunarayan,

K. (Eds.), *The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings*, Vol. 5318 of *Lecture Notes in Computer Science*, pp. 649–664. Springer.

Kuncak, V., & Rinard, M. C. (2007). Towards Efficient Satisfiability Checking for Boolean Algebra with Presburger Arithmetic. In Pfenning, F. (Ed.), *Automated Deduction - CADE-21, 21st International Conference on Automated Deduction, Bremen, Germany, July 17-20, 2007, Proceedings*, Vol. 4603 of *Lecture Notes in Computer Science*, pp. 215–230. Springer.

Kupke, C., Pattinson, D., & Schröder, L. (2022). Coalgebraic Reasoning with Global Assumptions in Arithmetic Modal Logics. *ACM Transactions on Computational Logic*, *23*(2).

Lutz, C. (2007). Inverse Roles Make Conjunctive Queries Hard. In *Proceedings of the 2007 International Workshop on Description Logics (DL2007), Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June, 2007*, Vol. 250 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Lutz, C. (2008a). The Complexity of Conjunctive Query Answering in Expressive Description Logics. In Armando, A., Baumgartner, P., & Dowek, G. (Eds.), *Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings*, Vol. 5195 of *Lecture Notes in Computer Science*, pp. 179–193. Springer.

Lutz, C. (2008b). Two Upper Bounds for Conjunctive Query Answering in SHIQ. In *Proceedings of the 21st International Workshop on Description Logics (DL2008), Dresden, Germany, May 13-16, 2008*, Vol. 353 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Ngo, N., Ortiz, M., & Simkus, M. (2016). Closed Predicates in Description Logics: Results on Combined Complexity. In Baral, C., Delgrande, J. P., & Wolter, F. (Eds.), *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*, pp. 237–246. AAAI Press.

Ortiz, M. (2010). *Query Answering in Expressive Description LLogics: Techniques and Complexity Results*. Ph.D. thesis, TU Wien, AT.

Ortiz, M., Rudolph, S., & Simkus, M. (2010). Worst-Case Optimal Reasoning for the Horn-DL Fragments of OWL 1 and 2. In Lin, F., Sattler, U., & Truszczynski, M. (Eds.), *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*. AAAI Press.

Ortiz, M., & Simkus, M. (2012). Reasoning and Query Answering in Description Logics. In Eiter, T., & Krennwallner, T. (Eds.), *Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School 2012, Vienna, Austria, September 3-8, 2012. Proceedings*, Vol. 7487 of *Lecture Notes in Computer Science*, pp. 1–53. Springer.

Ortiz, M., & Simkus, M. (2014). Revisiting the Hardness of Query Answering in Expressive Description Logics. In Kontchakov, R., & Mugnier, M. (Eds.), *Web Reasoning and*

*Rule Systems - 8th International Conference, RR 2014, Athens, Greece, September 15-17, 2014. Proceedings*, Vol. 8741 of *Lecture Notes in Computer Science*, pp. 216–223. Springer.

Ortiz, M., Simkus, M., & Eiter, T. (2008). Worst-case Optimal Conjunctive Query Answering for an Expressive Description Logic without Inverses. In Fox, D., & Gomes, C. P. (Eds.), *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pp. 504–510. AAAI Press.

Otto, M. (2004). Modal and Guarded Characterisation Theorems over Finite Transition Systems. *Annals of Pure and Applied Logic*, *130*(1-3), 173–205.

Peñaloza, R., & Potyka, N. (2017). Towards Statistical Reasoning in Description Logics over Finite Domains. In Moral, S., Pivert, O., Sánchez, D., & Marín, N. (Eds.), *Scalable Uncertainty Management - 11th International Conference, SUM 2017, Granada, Spain, October 4-6, 2017, Proceedings*, Vol. 10564 of *Lecture Notes in Computer Science*, pp. 280–294. Springer.

Piro, R. (2012). *Model-theoretic characterisations of description logics*. Ph.D. thesis, University of Liverpool, UK.

Schild, K. (1991). A Correspondence Theory for Terminological Logics: Preliminary Report. In Mylopoulos, J., & Reiter, R. (Eds.), *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991*, pp. 466–471. Morgan Kaufmann.