$\mathcal{ALC}_{\mathcal{ALC}}$: a Context Description Logic

Szymon Klarman[1] and Víctor Gutiérrez-Basulto[2]

[1] Department of Computer Science, Vrije Universiteit Amsterdam
sklarman@few.vu.nl
[2] Department of Computer Science, Universität Bremen
victor@informatik.uni-bremen.de

Abstract. We develop a novel description logic (DL) for representing and reasoning with contextual knowledge. Our approach descends from McCarthy's tradition of treating contexts as formal objects over which one can quantify and express first-order properties. As a foundation we consider several common product-like combinations of DLs with multi-modal logics and adopt the prominent $(\mathbf{K}_n)_{\mathcal{ALC}}$. We then extend it with a second sort of vocabulary for describing contexts, i.e., objects of the second dimension. In this way, we obtain a *two-sorted, two-dimensional combination of a pair of DLs* $\mathcal{ALC}$, called $\mathcal{ALC}_{\mathcal{ALC}}$. As our main technical result, we show that the satisfiability problem in this logic, as well as in its proper fragment $(\mathbf{K}_n)_{\mathcal{ALC}}$ with global TBoxes and local roles, is 2ExpTime-complete. Hence, the surprising conclusion is that the significant increase in the expressiveness of $\mathcal{ALC}_{\mathcal{ALC}}$ due to adding the vocabulary comes for no substantial price in terms of its worst-case complexity.

1 Introduction

Over two decades ago John McCarthy introduced the AI community to a new paradigm of formalizing contexts in logic-based knowledge systems. This idea, presented in his Turing Award Lecture [1], was quickly picked up by others and by now has led to a significant body of work studying different implementations of the approach in a variety of formal frameworks and applications [2,3,4,5,6,7,8]. The great appeal of McCarthy's paradigm stems from the simplicity and intuitiveness of the three major postulates it is based on:

1. Contexts are formal objects. More precisely, a *context* is anything that can be denoted by a first-order term and used meaningfully in a statement of the form $ist(c, p)$, saying that proposition p *is true* in context c [1,5,6,2], e.g., $ist(Hamlet,$ 'Hamlet is a prince.'$)$. By adopting a strictly formal view on contexts, one can bypass unproductive debates on what they really are and instead take them as primitives underlying practical models of contextual reasoning.

2. Contexts are organized in relational structures. In the commonsense reasoning, contextual assumptions are dynamically and directionally altered [8,2]. Contexts are entered and then exited, accessed from other contexts or transcended to broader ones. Formally, we want to allow nestings of the form $ist(c, ist(c', p))$, e.g., $ist(France, ist(capital,$ 'The city river is Seine.'$))$.

3. Contexts have properties and can be described. As first-order objects, contexts can be in a natural way described in a first-order language [4,6]. This allows for addressing them generically through quantified formulas such as $\forall x(P(x) \rightarrow ist(x,p))$, expressing that p is true in every context of type P, e.g., $\forall x(barbershop(x) \rightarrow ist(x,$ 'Main service is a haircut.'$))$.

The goal of this work is to import McCarthy's paradigm into the framework of Description Logics (DLs), a popular family of knowledge representation formalisms, with many successful applications [9]. Although the importance of contexts in DLs has been generally acknowledged, the framework is still not supported with a dedicated, generic theory of accommodating contextual knowledge. The most common perspectives considered in this area are limited to: 1) integration of local ontologies [10,11], 2) modeling levels of abstraction as subsets of DL models [12,13], and 3) capturing dynamics of knowledge across a fixed modal dimension, most typically a temporal one [14,15,16].

The DL $\mathcal{ALC}_{\mathcal{ALC}}$, which we develop here, is a novel formalism for representing and reasoning with context-dependent knowledge. On the one hand, we systematically incorporate the three postulates of McCarthy, and thus, ground our proposal in a longstanding tradition of formalizing contexts in AI. On the other, we build on top of two-dimensional DLs [17], which provide $\mathcal{ALC}_{\mathcal{ALC}}$ with well-understood formal foundations. In this paper we present a thorough study of the formal properties of $\mathcal{ALC}_{\mathcal{ALC}}$, including its expressiveness, computational complexity and relationships to other formalisms. As our main technical result, we show that the satisfiability problem in $\mathcal{ALC}_{\mathcal{ALC}}$, as well as in its proper fragment $(\mathbf{K}_n)_{\mathcal{ALC}}$ with global TBoxes and local roles, is 2ExpTime-complete. This reveals that the jump in the complexity from ExpTime is essentially caused by the interaction of multiple $\mathbf{K}$-modalities with global TBoxes.

2 Overview

We start with an outline of the milestones for constructing and studying the logic $\mathcal{ALC}_{\mathcal{ALC}}$. Then, we recap the basic notions concerning the DL $\mathcal{ALC}$.

2.1 Roadmap

We introduce $\mathcal{ALC}_{\mathcal{ALC}}$ in a gradual way. First, in Section 3, we elaborate on some well-studied combinations of the DL $\mathcal{ALC}$ with modal logics, known as two-dimensional or modal DLs [18,17,19]. From our perspective, the two-dimensional semantics of such logics is very well suited for representing context objects and the relational structures they form. After some conceptual and computational evaluation we then adopt $(\mathbf{K}_n)_{\mathcal{ALC}}$ as the foundation for our context DL. Finally, we show that the migration from $\mathcal{ALC}$ to $(\mathbf{K}_n)_{\mathcal{ALC}}$ with global TBoxes and local roles rises the complexity from ExpTime to 2ExpTime.

Next, in Section 4, we extend $(\mathbf{K}_n)_{\mathcal{ALC}}$ with a second sort of vocabulary, which serves for describing contexts. Formally, we can see this extension as a shift from $(\mathbf{K}_n)_{\mathcal{ALC}}$ to $\mathcal{ALC}_{\mathcal{ALC}}$, i.e., a *two-sorted*, *two-dimensional* combination of a pair of DLs $\mathcal{ALC}$. Each sort in $\mathcal{ALC}_{\mathcal{ALC}}$ applies to its corresponding dimension and the two are allowed to interact in a controlled manner. Since such an extension is relatively uncommon, we then relate $\mathcal{ALC}_{\mathcal{ALC}}$ to the standard framework of products of modal logics and show that the departure is not radical. More interestingly, we also prove that the extension, although offering a lot of expressive flexibility, is not to be paid for in yet another increase of the worst-case complexity. Satisfiability in $\mathcal{ALC}_{\mathcal{ALC}}$ remains 2ExpTime-complete.

In Section 5, we present an example application of $\mathcal{ALC}_{\mathcal{ALC}}$. Finally, in Section 6, we conclude the paper and point to directions for future research.

2.2 Preliminaries: DL $\mathcal{ALC}$

A DL language is specified by a vocabulary $\Sigma = (N_I, N_C, N_R)$, where N_I is a set of *individual names*, N_C a set of *concept names*, N_R a set of *role names*, and a number of operators for constructing complex concept descriptions [9]. The $\mathcal{ALC}$ concept language L over Σ is the smallest set of concepts containing $\top$, all concept names from N_C and closed under the constructors:

$$\neg C \mid C \sqcap D \mid \exists r.C$$

where $C, D \in L$ and $r \in N_R$. Conventionally, we abbreviate $\neg\top$ with $\bot$, $\neg(\neg C \sqcap \neg D)$ with $C \sqcup D$ and $\neg\exists r.\neg C$ with $\forall r.C$. The semantics of L is given through *interpretations* of the form $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, where Δ is a non-empty *domain* of individuals, and $\cdot^{\mathcal{I}}$ is an *interpretation function*. The meaning of the vocabulary is fixed via mappings: $a^{\mathcal{I}} \in \Delta$ for every $a \in N_I$, $A^{\mathcal{I}} \subseteq \Delta$ for every $A \in N_C$ and $r^{\mathcal{I}} \subseteq \Delta \times \Delta$ for every $r \in N_R$, and $\top^{\mathcal{I}} = \Delta$. Then the function is inductively extended over L according to the fixed semantics of the constructors:

$$\begin{aligned}(\neg C)^{\mathcal{I}} &= \{x \in \Delta \mid x \notin C^{\mathcal{I}}\},\\ (C \sqcap D)^{\mathcal{I}} &= \{x \in \Delta \mid x \in C^{\mathcal{I}} \cap D^{\mathcal{I}}\},\\ (\exists r.C)^{\mathcal{I}} &= \{x \in \Delta \mid \exists y :\ \langle x, y\rangle \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}.\end{aligned}$$

A *knowledge base* (or an *ontology*) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. The TBox contains general concept inclusion axioms (GCIs) $C \sqsubseteq D$, for arbitrary concepts $C, D \in L$. We write $C \equiv D$ whenever both $C \sqsubseteq D$ and $D \sqsubseteq C$ are in $\mathcal{T}$. The ABox consists of concept assertions $C(a)$ and role assertions $r(a, b)$, where $a, b \in N_I$, $C \in L$ and $r \in N_R$. An interpretation $\mathcal{I}$ *satisfies* an axiom in either of the following cases:

- $\mathcal{I} \models C \sqsubseteq D$ *iff* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$,
- $\mathcal{I} \models C(a)$ *iff* $a^{\mathcal{I}} \in C^{\mathcal{I}}$,
- $\mathcal{I} \models r(a, b)$ *iff* $\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in r^{\mathcal{I}}$.

Finally, $\mathcal{I}$ is a *model* of a DL knowledge base whenever it satisfies all its axioms.

3 Adding context structures: from $\mathcal{ALC}$ to $(\mathrm{K}_n)_{\mathcal{ALC}}$

In order to introduce *context structures* into the DL semantics, and thus account for the first two postulates of McCarthy, we move from $\mathcal{ALC}$ to its two-dimensional, multi-modal extensions.

3.1 Syntax and semantics

A two-dimensional, multi-modal concept language $L_{\mathcal{ALC}}$ over vocabulary Σ is the smallest set of concepts containing $\top$, concept names from N_C and closed under the $\mathcal{ALC}$ and the two new constructors:

$$\Diamond_i C \ \mid \ \Box_i C$$

where $C \in L_{\mathcal{ALC}}$ and $1 \leq i \leq n$ for some fixed $n \in \mathbb{N}$. It is assumed that $\Box_i$ abbreviates $\neg\Diamond_i\neg$. In our framework, every i is interpreted as a distinguished *contextualization operation*. The modal *context operators* associated with i enable a transition to the state of affairs holding in some ($\Diamond_i$) or all ($\Box_i$) contexts accessible from the current one through i. An interpretation of $L_{\mathcal{ALC}}$ is defined as a tuple $\mathfrak{M} = (\mathfrak{C}, \{R_i\}_{1 \leq i \leq n}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c \in \mathfrak{C}})$, where:

- $\mathfrak{C}$ is a non-empty *context domain*,
- $R_i \subseteq \mathfrak{C} \times \mathfrak{C}$ is an *accessibility relation* on $\mathfrak{C}$, associated with $\Diamond_i$ and $\Box_i$,
- Δ is a non-empty *object domain*,
- $\cdot^{\mathcal{I}(c)}$ is an *interpretation function* in context c.

For every $c \in \mathfrak{C}$, the interpretation function $\mathcal{I}(c)$ fixes the meaning of the language by extending the basic $\mathcal{ALC}$ interpretation rules with the additional:

$$(\Diamond_i C)^{\mathcal{I}(c)} = \{x \in \Delta \mid \exists d \in \mathfrak{C} : \ cR_i d \land x \in C^{\mathcal{I}(d)}\},$$
$$(\Box_i C)^{\mathcal{I}(c)} = \{x \in \Delta \mid \forall d \in \mathfrak{C} : \ cR_i d \rightarrow x \in C^{\mathcal{I}(d)}\}.$$

In what follows, we loosely refer to $\mathfrak{C}$ as the *context dimension* and to Δ as the *object dimension* of the combination (see example in Fig. 1). Generally, the semantic setup for multi-dimensional DLs allows several degrees of freedom regarding rigidity of names and domain assumptions [17]. Here, we pose the natural, rigid interpretation of individual names, i.e., $a^{\mathcal{I}(c)} = a^{\mathcal{I}(d)}$ for every $c, d \in \mathfrak{C}$, and local (non-rigid) interpretation of concepts. The interpretation of roles is discussed in the next paragraphs. We also assume that all contexts share the same object domain. Even if not suiting all applications, the constant domain assumption is known to be most universal, in the sense that the expanding/varying case can be always reduced to the constant one.

For a fixed language $L_{\mathcal{ALC}}$ the knowledge about the object dimension, now relative to contexts, can be expressed by means of usual axioms. In particular, a TBox $\mathcal{T}$ is a set of GCIs over concepts from $L_{\mathcal{ALC}}$. In this section it suffices to consider only the basic problem of *concept satisfiability* with respect to a global $\mathcal{T}$. The satisfaction relation for GCIs is defined with respect to an interpretation $\mathfrak{M}$ and a context $c \in \mathfrak{C}$:

- $(\mathfrak{M}, c) \models C \sqsubseteq D$ *iff* $C^{\mathcal{I}(c)} \subseteq D^{\mathcal{I}(c)}$.

We call $\mathfrak{M}$ a model of a global $\mathcal{T}$ whenever it satisfies all axioms in $\mathcal{T}$ in every $c \in \mathfrak{C}$. A concept C is satisfiable w.r.t. $\mathcal{T}$ *iff* there exists a model of $\mathcal{T}$ such that for some $c \in \mathfrak{C}$ and $d \in \Delta$ it is the case that $d \in C^{\mathcal{I}(c)}$.

It is not hard to see that without further constraints the resulting logic corresponds to the well-known product of multi-modal $\mathbf{K}_n$ with $\mathcal{ALC}$, denoted

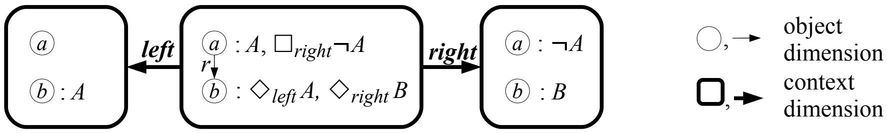

Fig. 1. A context structure modeling concept $A \sqcap \Box_{right} \neg A \sqcap \exists r.(\Diamond_{left} A \sqcap \Diamond_{right} B)$.

shortly as $(\mathbf{K}_n)_{\mathcal{ALC}}$ [18,20,17,19]. As for many other applications, also in the case of context DLs $(\mathbf{K}_n)_{\mathcal{ALC}}$ seems to provide the most natural and flexible foundation. Obviously, it is not difficult to further constrain accessibility relations in order to obtain context structures with more specific properties. Leaving a broader study of this subject for future research, let us just consider two such restrictions, sometimes evoked in the literature on contexts:

(quasi-functionality) $\forall c, d, e \in \mathfrak{C}\ (cRd \wedge cRe \rightarrow d = e)$,
(seriality) $\forall c \in \mathfrak{C}\ \exists d \in \mathfrak{C}\ (cRd)$.

Buvač's propositional logic of contexts [2,3] is a notational variant of $\mathbf{K}_n$, with $\Box_i \varphi$ written as $ist(i, \varphi)$. In Buvač's setting $\Box_i$ quantifies over possible interpretations of the context i. In our framework, where contexts are not modality indices but first-order objects, $\Box_i$ would quantify over possible contexts instead, which clearly distorts the intended behavior of *ist*. To avoid this, one might rather use $\Box_i$ of the logic $\mathbf{Alt}_n$, characterized by all quasi-functional Kripke frames [19]. In $\mathbf{Alt}_n$ there is at most one context accessible through each contextualization operation. Thus, $\Diamond_i \varphi \wedge \Diamond_i \psi$ semantically implies $ist(c, \varphi \wedge \psi)$ for some unique c. Nossum [8] pursues similar intuitions and advocates even stronger $\mathbf{DAlt}_n$, which is Kripke-complete w.r.t. all quasi-functional and serial frames. Such a semantics ensures that it is always possible to reach exactly one context through each accessibility relation. Since formally the two frame properties boil down to the functionality condition, it follows that the two operators $\Diamond_i, \Box_i$ collapse into a single $\bigcirc_i$. Finally $\mathbf{D}_n$, characterized by all serial frames, is used by Buvač [2,3] for verifying consistency of contextual knowledge. Since the seriality condition enforces existence of all potential contexts, the knowledge attributed to these contexts cannot be self-contradictory.

3.2 Complexity

As it turns out, the choice between any of the characterizations discussed above is quite irrelevant from the computational perspective. In most cases the complexity results apply to all logics $L_{\mathcal{ALC}}$, for $L \in \{\mathbf{DAlt}_n, \mathbf{D}_n, \mathbf{Alt}_n, \mathbf{K}_n\}$. To ease the transfer of some of the observations we make below, we use the following reductions:

Proposition 1. *Concept satisfiability w.r.t. global TBoxes is polynomially reducible between the following logics (where $\mapsto$ means* reduces to*):*

$$(\mathbf{DAlt}_n)_{\mathcal{ALC}} \mapsto \{(\mathbf{D}_n)_{\mathcal{ALC}}, (\mathbf{Alt}_n)_{\mathcal{ALC}}\} \mapsto (\mathbf{K}_n)_{\mathcal{ALC}}.$$

To see that the reductions hold indeed, it is enough to notice that if $(C, \mathcal{T})$ is a problem of deciding whether a concept C is satisfiable w.r.t. a global TBox $\mathcal{T}$, then by simple transformations of C and $\mathcal{T}$ one can enforce only models that are bisimilar to those characterizing the respective frame conditions:

(quasi-functionality) W.l.o.g. assume that $C = \mathrm{NNF}(C)$, where NNF stands for Negation Normal Form, and $\mathcal{T} = \{\top \sqsubseteq C_{\mathcal{T}}\}$, for some $C_{\mathcal{T}} = \mathrm{NNF}(C_{\mathcal{T}})$. Let C' and $C'_{\mathcal{T}}$ be the result of replacing every subconcept $\Diamond_i B$ occurring in C and $C_{\mathcal{T}}$, respectively, with $(\Diamond_i \top) \sqcap (\Box_i B)$. Then, $(C, \mathcal{T})$ is satisfiable on a quasi-functional frame *iff* $(C', \{\top \sqsubseteq C'_{\mathcal{T}}\})$ is satisfiable.

(seriality) Let $\mathcal{T}' = \mathcal{T} \cup \{\top \sqsubseteq \Diamond_i \top \mid 1 \leq i \leq n\}$, where n is the number of all modalities occurring in $\mathcal{T}$ and C. Then, $(C, \mathcal{T})$ is satisfiable on a serial frame *iff* $(C, \mathcal{T}')$ is satisfiable.

Our first result is a negative one. It closes the option of using rigid roles, i.e., such that $r^{\mathcal{I}(c)} = r^{\mathcal{I}(d)}$ for every $c, d \in \mathfrak{C}$, or applying context operators to roles. Unfortunately, adding rigid roles leads to undecidability already for the strongest of the logics with just a single context operator.

Theorem 1. *Concept satisfiability in* $\mathbf{DAlt}_{\mathcal{ALC}}$ *w.r.t. global TBoxes and with a single rigid role is undecidable.*

The full proof, along the others from this paper, is included in the appendix. We notice that $\mathbf{DAlt}_{\mathcal{ALC}}$ corresponds to a fragment of $\mathrm{LTL}_{\mathcal{ALC}}$ with the *next-time* operator, which is enough to construct a usual encoding of the undecidable $\mathbb{N} \times \mathbb{N}$ *tiling problem* [14]. Together with Proposition 1, the theorem immediately entails the following:

Theorem 2. *For any* $L \in \{\mathbf{DAlt}_n, \mathbf{D}_n, \mathbf{Alt}_n, \mathbf{K}_n\}$*, concept satisfiability in* $L_{\mathcal{ALC}}$ *w.r.t. global TBoxes with a single rigid role is undecidable.*

This result reveals an obvious limitation to the formalism, but a limitation one has to live with, considering that combinations of rigid roles with global TBoxes are rarely decidable unless the expressive power of the modal or the DL component is significantly reduced [19,14]. In the rest of this paper, we almost exclusively address the case of local (non-rigid) roles. To show decidability and the upper bound of the concept satisfiability problem in this setup,[3] we devise a quasistate elimination algorithm for $(\mathbf{K}_n)_{\mathcal{ALC}}$, similar to [19, Theorem 6.61]. As usual, the idea is to abstract from the domains $\mathfrak{C}$ and Δ and consider only a finite, in fact double exponential, number of quasistates which represent possible contexts inhabited by a finite number of possible types of individuals. Then, we iteratively eliminate all those that do not satisfy necessary conditions.

Theorem 3. *Deciding concept satisfiability in* $(\mathbf{K}_n)_{\mathcal{ALC}}$ *w.r.t. global TBoxes and only with local roles is in* 2ExpTime*.*

[3] Mind that the NExpTime-completeness result for concept satisfiability in $\mathbf{K}_{\mathcal{ALC}}$ [19, Theorem 15.15] applies to $\mathcal{ALC}$ with a single pair of $\mathbf{K}$ operators, full booleans on modalized formulas and no global TBoxes.

One could hope that at least some of the considered logics could be less complex than that. However, as the next theorem shows, this is not the case.

Theorem 4. *Deciding concept satisfiability in* $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ *w.r.t. global TBoxes and only with local roles is* 2EXPTIME*-hard.*

For the proof we use a reduction of the word problem for exponentially bounded Alternating Turing Machines, which is known to be 2EXPTIME-hard [21]. The increase in the complexity by one exponential, as compared to $\mathcal{ALC}$ alone (for which the problem is EXPTIME-complete [9]), is notable and quite surprising. It could be expected that without rigid roles the satisfiability problem can be straightforwardly reduced to satisfiability in fusion models. This in turn should yield EXPTIME upper bound by means of the standard techniques. However, as the following example for $(\mathbf{K}_n)_{\mathcal{ALC}}$ demonstrates, this strategy fails.

$$(\dagger)\ \Diamond_i C \sqcap \exists r.\Box_i \bot \qquad (\ddagger)\ \exists succ_i.C \sqcap \exists r.\forall succ_i.\bot$$

Although (†) clearly does not have a model, its reduction (‡) to a fusion language, where context operators are translated to restrictions on fresh $\mathcal{ALC}$ roles, is satisfiable. The reason is that while in the former case the information about the structure of the **K**-frame is global for all individuals, in the latter it becomes local. The r-successor in (‡) is simply not 'aware' that it should actually have a $succ_i$-successor.[4] This effect, amplified by presence of multiple modalities and global TBoxes (which can enforce infinite **K**-trees), makes the reasoning harder.

The two complexity bounds from Theorem 3 and 4, together with the reductions established in Proposition 1, provide us with the completeness result.

Theorem 5. *For any* $L \in \{\mathbf{DAlt}_n, \mathbf{D}_n, \mathbf{Alt}_n, \mathbf{K}_n\}$*, deciding concept satisfiability in* $L_{\mathcal{ALC}}$ *w.r.t. global TBoxes and only with local roles is* 2EXPTIME*-complete.*

The theorem is quite robust under changes of domain assumptions and holds already in the case of expanding/varying domains in $(\mathbf{Alt}_n)_{\mathcal{ALC}}$. The only exception applies to $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ and $(\mathbf{D}_n)_{\mathcal{ALC}}$ with expanding/varying domains, where reduction to $\mathcal{ALC}$ is still possible.

What follows from this analysis, is that by sacrificing the generality of $\mathbf{K}_n$-frames one does not immediately obtain a better computational behavior as long as multiple context operators are permitted. For this reason, we adopt $(\mathbf{K}_n)_{\mathcal{ALC}}$ as the baseline for $\mathcal{ALC}_{\mathcal{ALC}}$, leaving for now the option of restricting context structures as an open problem.

4 Describing contexts: from $(\mathbf{K}_n)_{\mathcal{ALC}}$ to $\mathcal{ALC}_{\mathcal{ALC}}$

We are now ready to define the target logic $\mathcal{ALC}_{\mathcal{ALC}}$, which additionally to $(\mathbf{K}_n)_{\mathcal{ALC}}$ offers a second sort of vocabulary for directly describing contexts. This extension addresses the third postulate of McCarthy.

[4] Demonstrating the corresponding phenomenon in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ is not that straightforward due to the seriality condition, as then the global information concerns only the existence of $succ_i$-predecessors. Thus, one needs role inverses in the fusion language to observe the loss of such information.

4.1 Syntax and semantics

We start by introducing the context component of the language and then suitably revise the object component.

The *context language* L_C is an $\mathcal{ALC}$ concept language over vocabulary $\Gamma = (M_I, M_C, M_R)$, where M_I is a set of (*context*) *individual names*, M_C is a set of (*context*) *concept names*, and M_R is a set of (*context*) *role names*. For disambiguation, we use ***bold font*** when writing names from the context vocabulary and we denote the elements of L_C as *c-concepts*. The semantics is defined in the usual manner (as presented in Section 2.2), in terms of an interpretation function $\cdot^{\mathcal{J}}$ ranging over the context domain $\mathfrak{C}$. The *context knowledge base* $\mathcal{C}$ consists of TBox and ABox axioms over Γ and L_C, also with the usual satisfaction conditions. Thus, $\mathcal{C}$ is in fact a standard $\mathcal{ALC}$ ontology with standard models of the form $(\mathfrak{C}, \cdot^{\mathcal{J}})$.

The interpretations of the context language are incorporated in the full $\mathcal{ALC}_{\mathcal{ALC}}$ interpretations of the form $\mathfrak{M} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c \in \mathfrak{C}})$, where:

- $\mathfrak{C}$ is a non-empty *context domain*,
- $\cdot^{\mathcal{J}}$ is an *interpretation function* of the context language,
- Δ is a non-empty *object domain*,
- $\cdot^{\mathcal{I}(c)}$ is an *interpretation function* of the object language in c.

The divergence from the original $(\mathbf{K}_n)_{\mathcal{ALC}}$ interpretations is minor. Basically, the accessibility relations over $\mathfrak{C}$ become now redundant, as their function can be taken over by context roles. For every contextualization operation i we can assume an implicit correspondence $R_i = \boldsymbol{r}_i^{\mathcal{J}}$, for some $\boldsymbol{r}_i \in M_R$. Note that given the broadened take on the context dimension, we might be now less strict about the informal reading of some of the components of the framework. Arguably, not all context roles have to be necessarily seen as 'contextualization operations' and not all elements of $\mathfrak{C}$ as genuine 'contexts'. Sometimes they can be just entities needed for describing contexts. Nevertheless, we keep using the context-object nomenclature to avoid potential confusions.

Although one can already express rich knowledge about contexts, such knowledge remains 'invisible' from the object level. In order to render it more accessible, and so gain better control over the interaction between the dimensions, we need to suitably internalize context descriptions in the object language.

Let $\Sigma = (N_I, N_C, N_R)$ be the *object vocabulary* disjoint from Γ. The *object language* L_O over Σ and the context language L_C is the smallest set of concepts, called *o-concepts*, containing $\top$, concept names from N_C and closed under the $\mathcal{ALC}$ and the following two constructors:

$$\langle \boldsymbol{C} \rangle_{\boldsymbol{r}} D \ \mid \ [\boldsymbol{C}]_{\boldsymbol{r}} D$$

where $\boldsymbol{C} \in L_C$ and $\boldsymbol{r} \in M_R$. Again, $[\cdot]_{\boldsymbol{r}}$ abbreviates $\neg\langle \cdot \rangle_{\boldsymbol{r}} \neg$. Intuitively, $\langle \boldsymbol{C} \rangle_{\boldsymbol{r}} D$ denotes all objects which are D in *some* context which is $\boldsymbol{C}$ and is accessible through $\boldsymbol{r}$. Similarly, $[\boldsymbol{C}]_{\boldsymbol{r}} D$ denotes all objects which are D in *every* context which is $\boldsymbol{C}$ and is accessible through $\boldsymbol{r}$. Overall, the syntax of the object language diverges from the one of $(\mathbf{K}_n)_{\mathcal{ALC}}$ only in that the indices appearing by $\Diamond_i, \Box_i$

are now replaced with context roles, while both operators embrace a single c-concept, which additionally qualifies the accessed contexts. Consequently, the changes in the semantics affect only the contextualized concepts:

$$(\langle \boldsymbol{C}\rangle_{\boldsymbol{r}} D)^{\mathcal{I}(c)} = \{x \in \Delta \mid \exists d \in \mathfrak{C} : \ \langle c, d\rangle \in \boldsymbol{r}^{\mathcal{J}} \wedge d \in \boldsymbol{C}^{\mathcal{J}} \wedge x \in D^{\mathcal{I}(d)}\},$$
$$([\boldsymbol{C}]_{\boldsymbol{r}} D)^{\mathcal{I}(c)} = \{x \in \Delta \mid \forall d \in \mathfrak{C} : \ \langle c, d\rangle \in \boldsymbol{r}^{\mathcal{J}} \wedge d \in \boldsymbol{C}^{\mathcal{J}} \rightarrow x \in D^{\mathcal{I}(d)}\}.$$

To grant maximum flexibility in expressing the knowledge about the object dimension we first define the set of possible *object formulas*, i.e., formulas which can meaningfully hold in individual contexts:

$$B \sqsubseteq D \ \mid \ a : D \ \mid \ s(a, b) \ \mid \ \neg\varphi \ \mid \ \varphi \wedge \psi \ \mid \ \langle \boldsymbol{C}\rangle_{\boldsymbol{r}}\varphi \ \mid \ [\boldsymbol{C}]_{\boldsymbol{r}}\varphi$$

where B, D are o-concepts, $a, b \in N_I$, $s \in N_R$, $\boldsymbol{C}$ is a c-concept and $\boldsymbol{r} \in M_R$. Object formulas are satisfied by $\mathfrak{M}$ in context $c \in \mathfrak{C}$ in the following cases:

- $(\mathfrak{M}, c) \models B \sqsubseteq D$ *iff* $B^{\mathcal{I}(c)} \subseteq D^{\mathcal{I}(c)}$,
- $(\mathfrak{M}, c) \models a : D$ *iff* $a^{\mathcal{I}(c)} \in D^{\mathcal{I}(c)}$,
- $(\mathfrak{M}, c) \models s(a, b)$ *iff* $\langle a^{\mathcal{I}(c)}, b^{\mathcal{I}(c)}\rangle \in s^{\mathcal{I}(c)}$,
- $(\mathfrak{M}, c) \models \neg\varphi$ *iff* $(\mathfrak{M}, c) \not\models \varphi$,
- $(\mathfrak{M}, c) \models \varphi \wedge \psi$ *iff* $(\mathfrak{M}, c) \models \varphi$ and $(\mathfrak{M}, c) \models \psi$,
- $(\mathfrak{M}, c) \models \langle \boldsymbol{C}\rangle_{\boldsymbol{r}}\varphi$ *iff* $(\mathfrak{M}, d) \models \varphi$ for some $d \in \mathfrak{C}$ s.t. $\langle c, d\rangle \in \boldsymbol{r}^{\mathcal{J}}$ and $d \in \boldsymbol{C}^{\mathcal{J}}$,
- $(\mathfrak{M}, c) \models [\boldsymbol{C}]_{\boldsymbol{r}}\varphi$ *iff* $(\mathfrak{M}, d) \models \varphi$ for every $d \in \mathfrak{C}$ s.t. $\langle c, d\rangle \in \boldsymbol{r}^{\mathcal{J}}$ and $d \in \boldsymbol{C}^{\mathcal{J}}$.

Then we define an *object knowledge base* $\mathcal{O}$ as a set of axioms of two forms:

$$\boldsymbol{a} : \varphi \ \mid \ \boldsymbol{C} : \varphi$$

where $\boldsymbol{a} \in M_I$, $\boldsymbol{C}$ is a c-concept and φ is an object formula. Such axioms have a straightforward reading: φ is true in context $\boldsymbol{a}$; and φ is true in every context which is $\boldsymbol{C}$. Formally, we specify those conditions as follows:

- $\mathfrak{M} \models \boldsymbol{a} : \varphi$ *iff* $(\mathfrak{M}, c) \models \varphi$ for $c = \boldsymbol{a}^{\mathcal{J}}$,
- $\mathfrak{M} \models \boldsymbol{C} : \varphi$ *iff* $(\mathfrak{M}, c) \models \varphi$ for every $c \in \boldsymbol{C}^{\mathcal{J}}$.

A pair $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ is called an $\mathcal{ALC}_{\mathcal{ALC}}$ *knowledge base*. An interpretation $\mathfrak{M}$ is a *model* of $\mathcal{K}$ whenever all axioms in $\mathcal{K}$ are satisfied. A small example of an $\mathcal{ALC}_{\mathcal{ALC}}$ knowledge base is presented in Section 5.

4.2 Complexity and expressiveness

Obviously, the expressiveness of $\mathcal{ALC}_{\mathcal{ALC}}$ properly subsumes that of $(\mathbf{K}_n)_{\mathcal{ALC}}$. In particular, the following relationship holds:

Proposition 2. *Concept satisfiability problem in $(\mathbf{K}_n)_{\mathcal{ALC}}$ w.r.t. global TBoxes is polynomially reducible to knowledge base satisfiability in $\mathcal{ALC}_{\mathcal{ALC}}$.*

To see this is indeed the case suppose $(C, \mathcal{T})$ is the problem of deciding whether concept C is satisfiable w.r.t. global TBox $\mathcal{T}$. Let C' and $\mathcal{T}'$ be the results of replacing every $\Diamond_i$ with $\langle\top\rangle_{\boldsymbol{r}_i}$ and every $\Box_i$ with $[\top]_{\boldsymbol{r}_i}$ in C and $\mathcal{T}$, respectively, where for $i \neq j$ we have $\boldsymbol{r}_i \neq \boldsymbol{r}_j$. Further define $\mathcal{C} = \emptyset$ and $\mathcal{O} = \{\boldsymbol{c} : a : C'\} \cup \{\top : C \sqsubseteq D \mid C \sqsubseteq D \in \mathcal{T}'\}$. It clearly follows that C is satisfiable w.r.t. $\mathcal{T}$ in $(\mathbf{K}_n)_{\mathcal{ALC}}$ *iff* the knowledge base $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ is satisfiable in $\mathcal{ALC}_{\mathcal{ALC}}$. Note, that the reduction holds even when object roles are interpreted rigidly.

This naturally means that the 2ExpTime lower bound established in Theorem 5 transfers immediately to $\mathcal{ALC}_{\mathcal{ALC}}$. But can it get even higher? Quite surprisingly, the answer is negative. Despite the increase of expressiveness, satisfiability problem in $\mathcal{ALC}_{\mathcal{ALC}}$ remains in 2ExpTime.

Theorem 6. *Deciding satisfiability of an $\mathcal{ALC}_{\mathcal{ALC}}$ knowledge base in which object roles are interpreted locally is* 2ExpTime-*complete.*

The proof of the upper bound is based on quasimodel elimination technique, which extends the one used for Theorem 3. In particular, every quasistate has to carry now also the type of the context which it represents and the set of object formulas which are satisfied in it.

To give a final insight into the expressiveness of the formalism, in more traditional terms of products of modal logics, we show that $\mathcal{ALC}_{\mathcal{ALC}}$ (with rigid roles) is equally expressive to the full $\mathcal{ALC}$ language over the union of two vocabularies interpreted in product models.

Let L_1 and L_2 be two $\mathcal{ALC}$ concept languages over disjoint vocabularies $\Gamma = (M_C, M_R, \emptyset)$ and $\Sigma = (N_C, N_R, \emptyset)$, respectively. Now, let $L_{1\times2}$ be the $\mathcal{ALC}$ concept language over vocabulary $\Theta = (M_C \cup N_C, M_R \cup N_R, \emptyset)$. The semantics for $L_{1\times2}$ is given through *product interpretations* $\mathcal{P} = (\mathfrak{C} \times \Delta, \cdot^{\mathcal{P}})$, which align every $r \in N_R$ along the 'vertical' dimension and every $\boldsymbol{p} \in M_R$ along the 'horizontal' one. Thus, $r^{\mathcal{P}}, \boldsymbol{p}^{\mathcal{P}} \subseteq (\mathfrak{C} \times \Delta) \times (\mathfrak{C} \times \Delta)$ and for every $u, v, w \in \mathfrak{C}$ and $x, y, z \in \Delta$:

$$\langle (u,x),(v,y)\rangle \in r^{\mathcal{P}} \to u = v \;\&\; \langle (w,x),(w,y)\rangle \in r^{\mathcal{P}},$$
$$\langle (u,x),(v,y)\rangle \in \boldsymbol{p}^{\mathcal{P}} \to x = y \;\&\; \langle (u,z),(v,z)\rangle \in \boldsymbol{p}^{\mathcal{P}}.$$

All concepts are interpreted as subsets of $\mathfrak{C} \times \Delta$. Additionally, we force every $\boldsymbol{A} \in M_C$ to be interpreted rigidly across the 'vertical' dimension, i.e., for every $v \in \mathfrak{C}$ and $x, y \in \Delta$ we assume:

$$(*) \quad (v,x) \in \boldsymbol{A}^{\mathcal{I}} \to (v,y) \in \boldsymbol{A}^{\mathcal{I}}$$

Finally, $\cdot^{\mathcal{P}}$ is extended inductively as usual. A concept $C \in L_{1\times2}$ is satisfiable *iff* for some product model $\mathcal{P} = (\mathfrak{C} \times \Delta, \cdot^{\mathcal{P}})$ it is the case that $C^{\mathcal{P}} \neq \emptyset$. On the contrary to the others, the condition $(*)$ is rather uncommon in the realm of products of modal logics. Nevertheless, it captures precisely the difference between the semantics of the two sorts of concepts. Without it the sorts collapse into one, while the whole logic turns into a notational variant of $(\mathbf{K}_n)_{\mathcal{ALC}}$. It turns out that the following claim holds:

Theorem 7. *The language $L_{1\times2}$ interpreted in product models is exactly as expressive as the concept language of $\mathcal{ALC}_{\mathcal{ALC}}$ interpreted in models with rigid interpretations of object roles.*

What follows from Theorem 7 is that the syntactic constraints of $\mathcal{ALC}_{\mathcal{ALC}}$, which make the logic more intuitive and well-behaved, by no means lead to loss of expressiveness. Moreover, it shows that $\mathcal{ALC}_{\mathcal{ALC}}$ (at least in its concept component) does not seriously deviate from the usual products of modal logics. In principle, the only feature distinguishing it from $(\mathbf{K}_n)_{\mathcal{ALC}}$ (both with and without rigid roles) is the condition (*) imposed on the interpretations of selected concepts, which in $\mathcal{ALC}_{\mathcal{ALC}}$ we simply happen to call context concepts.

5 Contextual ontologies — example

One of the designated applications of $\mathcal{ALC}_{\mathcal{ALC}}$ is construction of *contextual ontologies*. The distinguishing feature of such ontologies is that they allow for varying the characterization of concepts according to contexts. Hence, $\mathcal{ALC}_{\mathcal{ALC}}$ can provide a good formal support for exchanging and integrating information in DL. Moreover, as the context knowledge base can be created independently from the object component, the framework encourages reuse of existing ontologies.

As an example of a contextual ontology, we present a simple representation of knowledge about the food domain contextualized with respect to geographic locations. Consider the (context) geographic knowledge base $\mathcal{C} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a TBox and $\mathcal{A}$ an ABox.

$$\begin{aligned}
\mathcal{T} = \{\ & (1)\, \mathbf{Country} \sqsubseteq \exists \mathbf{location}.\mathbf{Europe} \sqcup \exists \mathbf{location}.\mathbf{America}\ ,\\
& (2)\, \mathbf{Region} \sqsubseteq \exists \mathbf{part_of}.\mathbf{Country},\\
& (3)\, \mathbf{City} \sqsubseteq \exists \mathbf{has_part}.\mathbf{Neighborhood}\ \}\\
\mathcal{A} = \{\ & (4)\, \mathit{US} : \mathbf{Country},\\
& (5)\, \mathit{SanFrancisco} : \mathbf{City},\\
& (6)\, \mathit{California} : \mathbf{Region},\\
& (7)\, \mathbf{part_of}(\mathit{California}, \mathit{US}),\\
& (8)\, \mathit{France} : \mathbf{Country} \sqcap \exists \mathbf{location}.\mathbf{Europe}\ \}
\end{aligned}$$

Now, we define an (object) food ontology $\mathcal{O}$, contextualized with $\mathcal{C}$.

$$\begin{aligned}
\mathcal{O} = \{\ & (a)\, \top : \mathsf{Food} \equiv \mathsf{Meat} \sqcup \mathsf{Beverages} \sqcup \mathsf{Sea_Food} \sqcup \mathsf{Grains}\\
& (b)\, \top : \mathsf{Wine} \equiv \mathsf{WhiteWine} \sqcup \mathsf{RedWine}\\
& (c)\, \top : (\mathtt{SauvignonBlanc} : \mathsf{WhiteWine})\\
& (d)\, \mathbf{Country} : [\mathbf{Europe}]_{\mathbf{location}}(\mathsf{WhiteWine} \sqsubseteq \mathsf{Popular_Beverage})\\
& (e)\, \mathit{California} : \mathsf{WhiteWine} \sqsubseteq [\mathbf{Country}]_{\mathbf{part_of}}\mathsf{Popular_Wine}\\
& (f)\, \mathit{US} : \mathsf{Popular_Wine} \sqsubseteq \neg\mathsf{Popular_Beverage}\\
& (g)\, \mathit{SanFrancisco} : [\top]_{\mathbf{has_part}}(\mathsf{WhiteWine} \sqsubseteq \neg\mathsf{Popular_Wine})\ \}
\end{aligned}$$

Let us shortly highlight the intuition behind $\mathcal{O}$ by explaining some of the axiom definitions and the inferences they sanction. First, axioms (a)-(c) present geographic-independent terminology of the food domain. For example, by (c), `SauvignonBlanc` is a WhiteWine in any part of the world. Then, (d)-(g) characterize WhiteWine as Popular_Wine or Popular_Beverage according to different territories. We explain (d)-(g) in terms of `SauvignonBlanc`. By (d), in any **Country** that has as a **location Europe** (e.g., *France*) `SauvignonBlanc` is a

Popular_Beverage. However, by (e)-(f), SauvignonBlanc is *not* a Popular_Beverage in *US*. This, is explained as follows: (e) establishes that SauvignonBlanc is a Popular_Wine in any **Country** of which *California* is part of, namely *US*. Then, by (f), in the *US* any Popular_Wine is not a Popular_Beverage. Hence, SauvignonBlanc is not a Popular_Beverage in *US*. Although SauvignonBlanc is a Popular_Wine in *US*, this is does not necessarily transfer to more specific contexts. For instance, by (g), in every part of *SanFrancisco*, SauvignonBlanc is not in fact a Popular_Wine. In particular, by (3), there is at least one such **Neighborhood** in which this happens.

6 Conclusions and future work

We have presented a novel DL $\mathcal{ALC}_{\mathcal{ALC}}$ for representing and reasoning with contextual knowledge. Our approach is derived from McCarthy's conception of contexts as first-order objects which are describable in a first-order language. Formally, the logic extends the well-known $(\mathbf{K}_n)_{\mathcal{ALC}}$ with another sort of 'context' vocabulary interpreted over the **K**-dimension. The surprising conclusion is that the increase of the expressiveness of the logic due to this addition comes for no substantial price in terms of the worst-case complexity. The jump to 2ExpTime-completeness stems from the interaction of multiple modalities with global TBoxes and is inherent already to the underlying two-dimensional DLs.

We believe that with this work we have set the stage for a promising future research on similar combinations of DLs. Clearly, there are three major determinants of such formalisms which deserve a careful study: 1) *the expressiveness of the context language*, 2) *the expressiveness of the object language*, 3) *the level of interaction between the two*. Finding a proper balance between them is the key to identifying well-behaved and potentially useful fragments. One of the first directions, which we want to investigate, is to reduce the interaction between the languages by employing only **S5**-like operators. Such operators, e.g., $\langle \boldsymbol{C} \rangle\varphi$, would state that there exists a context of type C in which φ holds, without involving context roles. This modification should result in a better computational behavior and a somewhat simpler conceptual design of the language.

On the applied side, it could be interesting to consider a restricted fragment of the framework (a finite number of named contexts) for the task of ontology integration on the Semantic Web. Arguably, such fragment is sufficient to provide a logical underpinning for the ongoing endeavor of describing and linking OWL/RDFS knowledge sources in a context-sensitive manner.

Acknowledgements We want to thank Carsten Lutz and Stefan Schlobach for many helpful discussions and suggestions on the ideas presented in this paper.

References

1. McCarthy, J.: Generality in artificial intelligence. Communications of the ACM **30** (1987) 1030–1035

2. Buvač, S., Mason, I.A.: Propositional logic of context. In: In Proceedings of the Eleventh National Conference on Artificial Intelligence. (1993) 412–419
3. Buvač, S., Buvac, V., Mason, I.A.: Metamathematics of contexts. Fundamenta Informaticae **23** 412–419
4. Buvač, S.: Quantificational logic of context. In: In Proceedings of the Eleventh National Conference on Artificial Intelligence. (1996) 412–419
5. McCarthy, J.: Notes on formalizing context. In: Proc. of International Joint Conference on Artificial Intelligence IJCAI93, Morgan Kaufmann (1993) 555–560
6. Guha, R.: Contexts: a formalization and some applications. PhD thesis, Stanford University (1991)
7. Guha, R., Mccool, R., Fikes, R.: Contexts for the semantic web. In: Proc. of the International Semantic Web Conference, Springer (2004) 32–46
8. Nossum, R.: A decidable multi-modal logic of context. Journal of Applied Logic **1**(1-2) (2003) 119 – 133
9. Baader, F., Calvanese, D., Mcguinness, D.L., Nardi, D., Patel-Schneider, P.F.: The description logic handbook: theory, implementation, and applications. Cambridge University Press (2003)
10. Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources. Journal of Data Semantics **1** (2003) 2003
11. Cuenca Grau, B., Kutz, O.: Modular ontology languages revisited. In: Proc. of the Workshop on Semantic Web for Collaborative Knowledge Acquisition. (2007)
12. Goczyla, K., Waloszek, W., Waloszek, A.: Contextualization of a DL knowledge base. In: The Proceedings of the International Workshop on Description Logics (DL-2007). (2007)
13. Grossi, D.: Desigining Invisible Handcuffs. Formal Investigations in Institutions and Organizations for Multi-Agent Systems. PhD thesis, Utrecht University (2007)
14. Lutz, C., Wolter, F., Zakharyaschev, M.: Temporal description logics: A survey. In: Proceedings of the Fourteenth International Symposium on Temporal Representation and Reasoning, IEEE Computer Society Press (2008)
15. Artale, A., Lutz, C., Toman, D.: A description logic of change. In Veloso, M., ed.: Proceedings of IJCAI'07. (2007) 218–223
16. Artale, A., Kontchakov, R., Lutz, C., Wolter, F., Zakharyaschev, M.: Temporalising tractable description logics. In: Proceedings of the Fourteenth International Symposium on Temporal Representation and Reasoning. (2007)
17. Wolter, F., Zakharyaschev, M.: Multi-dimensional description logics. In: IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, San Francisco, CA, USA (1999) 104–109
18. Baader, F., Laux, A.: Terminological logics with modal operators. In Mellish, C., ed.: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montréal, Canada, Morgan Kaufmann (1995) 808–814
19. Kurucz, A., Wolter, F., Zakharyaschev, M., Gabbay, D.M.: Many-Dimensional Modal Logics: Theory and Applications. Number 148 in Studies in Logic and the Foundations of Mathematics. Elsevier (2003)
20. Wolter, F., Zakharyaschev, M.: Satisfiability problem in description logics with modal operators. In: In Proceedings of the Sixth Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufman (1998) 512–523
21. Chandra, A.K., Kozen, D.C., Stockmeyer, L.J.: Alternation. J. ACM **28**(1) (1981) 114–133

Appendix

Theorem 1. *Concept satisfiability in* $\mathbf{DAlt}_{\mathcal{ALC}}$ *w.r.t. global TBoxes and with a single rigid role is undecidable.*

First, we observe the following correspondence:

Proposition 3. *A concept* C *is satisfiable w.r.t. a global TBox* $\mathcal{T}$ *in* $\mathbf{DAlt}_{\mathcal{ALC}}$ iff *it is satisfied w.r.t.* $\mathcal{T}$ *in some model* $\mathfrak{M} = (\mathbb{N}, <, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\mathbb{N}})$*, where* $\langle \mathbb{N}, < \rangle$ *is a linear order over natural numbers and* $<$ *is the accessibility relation of* $\bigcirc$*.*

Consequently, we can consider only such linear $\mathbf{DAlt}_{\mathcal{ALC}}$-models. This shows that $\mathbf{DAlt}_{\mathcal{ALC}}$ can be in fact seen as the subset of $\mathrm{LTL}_{\mathcal{ALC}}$ consisting of the $\mathcal{ALC}$ component and the *next-time* operator. This turns out to be enough to encode the undecidable $\mathbb{N} \times \mathbb{N}$ *tiling problem*, in the same way as in [14, Theorem 4]. An instance of the problem is defined as follows: given a finite set $S = \{t_0, \ldots, t_n\}$ of tile types, where each t_i is a 4-tuple of colors $\langle \mathit{left}(t_i), \mathit{right}(t_i), \mathit{up}(t_i), \mathit{down}(t_i)\rangle$, decide whether it is possible to cover $\mathbb{N} \times \mathbb{N}$-grid with tiles of these types. Moreover, it has to be ensured that only types of matching colors can be horizontal (vertical) neighbors in the tiling, i.e., ones for which $\mathit{right}(t_i) = \mathit{left}(t_j)$ ($\mathit{up}(t_i) = \mathit{down}(t_j)$). Let $A_0, \ldots, A_n$ be concept names representing the tile types from S and r be a rigid role. The following TBox $\mathcal{T}$ encodes the constraints of the tiling problem:

$$\top \sqsubseteq (\bigsqcup_{i\leq n} A_i) \sqcap (\bigsqcap_{i\neq j\leq n} \neg(A_i \sqcap A_j)) \tag{1}$$

$$\top \sqsubseteq \exists r.\top \tag{2}$$

$$A_i \sqsubseteq \forall r. \bigsqcup_{up(t_i)=down(t_j)} A_j, \text{ for every } \quad i \leq n \tag{3}$$

$$A_i \sqsubseteq \bigcirc \bigsqcup_{right(t_i)=left(t_j)} A_j, \text{ for every } \quad i \leq n \tag{4}$$

Now we can prove the target claim:

Lemma 1. *The concept* $\top$ *is satisfiable w.r.t.* $\mathcal{T}$ iff *there exists a tiling* $\tau : \mathbb{N} \times \mathbb{N} \mapsto S$.

Proof. ($\Rightarrow$) Let $\mathfrak{M} = (\mathbb{N}, <, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\mathbb{N}})$ be a model of $\mathcal{T}$. For an arbitrary individual $d \in \Delta$ we first fix the vertical axis ρ of the $\mathbb{N} \times \mathbb{N}$-grid:

- $\rho(0) = d$;
- $\rho(n + 1) = e$, for any e such that $\langle \rho(n), e\rangle \in r^{\mathcal{I}(0)}$.

By the axiom (2) of $\mathcal{T}$, every individual in the domain has an r-successor, hence, it is easy to see that the infinite chain ρ can be extracted from the model. Moreover by (1) it follows that every individual satisfy exactly one of the concepts representing tile types.

– $\tau(n,m) = t_i$ iff $\rho(m) \in A_i^{\mathcal{I}(n)}$

Finally, since r is rigid the conditions (3) and (4) of the encoding sufficiently guarantee proper coloring of the neighbors.

($\Leftarrow$) For tiling τ define an interpretation $\mathfrak{M} = (\mathbb{N}, <, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\mathbb{N}})$, where $\Delta = \{d_i \mid i \in \mathbb{N}\}$, and:

– $d_m \in A_i^{\mathcal{I}(n)}$ iff $\tau(n,m)$;
– $\langle d_n, d_{n+1}\rangle \in r^{\mathcal{I}(m)}$, for $n, m \in \mathbb{N}$.

Clearly $\top$ and all axioms from $\mathcal{T}$ are satisfied by $\mathcal{I}$ so we obtain a desired $\mathbf{DAlt}_{\mathcal{ALC}}$ model. ❑

Theorem 3. *Deciding concept satisfiability in* $(\mathbf{K}_n)_{\mathcal{ALC}}$ *w.r.t. global TBoxes and only with local roles is in* 2ExpTime.

Let $(C, \mathcal{T})$ be a problem of deciding whether a concept C is satisfiable w.r.t. the global TBox $\mathcal{T}$ in $(\mathbf{K}_n)_{\mathcal{ALC}}$. We devise a quasistate elimination algorithm which provides a correct answer in at most double exponential time w.r.t. the size of $(C, \mathcal{T})$. W.l.o.g. we assume that $\mathcal{T}$ is given as a single axiom $\top \sqsubseteq C_{\mathcal{T}}$, where $C_{\mathcal{T}} = \bigsqcap_{C \sqsubseteq D \in \mathcal{T}} \neg C \sqcup D$ and that all occurrences of $\Box_i$ in both C and $C_{\mathcal{T}}$ are replaced with $\neg\Diamond_i\neg$, while all occurrences of $\forall r.$ with $\neg\exists r.\neg$. We write $con(C, \mathcal{T})$ to denote the set of all subconcepts of C and $C_{\mathcal{T}}$, closed under negation. Similarly, by $rol(C, \mathcal{T})$ we denote the set of all role names occurring in C and $C_{\mathcal{T}}$.

A type for C and $\mathcal{T}$ is a subset $t \subseteq con(C, \mathcal{T})$ satisfying the following conditions:

– $A \in t$ iff $\neg A \notin t$, for all $A \in con(C, \mathcal{T})$,
– $C \sqcap D \in t$ iff $\{C, D\} \subseteq t$, for all $C \sqcap D \in con(C, \mathcal{T})$,
– $C_{\mathcal{T}} \in t$.

Let $\prod(C, \mathcal{T})$ be the set of all types for C and $\mathcal{T}$. We say that $t, t' \in \prod(C, \mathcal{T})$ are *k-compatible*, for $k \in (1, n)$, *iff* $\{\neg C \mid \neg\Diamond_k C \in t\} \subseteq t'$. A *quasistate* for $(C, \mathcal{T})$ is a subset of types $q \subseteq \prod(C, \mathcal{T})$, such that for every $t \in q$ and every $\exists r.D \in con(C, \mathcal{T})$:

(QS:) if $\exists r.D \in t$ then there is a type $t' \in q$ s.t. $\{D\} \cup \{\neg C \mid \neg\exists r.C \in t\} \subseteq t'$.

Two quasistates q and q' are *k-compatible*, for $k \in (1, n)$, *iff* there exists a pair of functions $f : q \mapsto q'$ and $g : q' \mapsto q$ such that:

– for every $t \in q$, t and $f(t)$ are k-compatible,
– for every $t \in q'$, $g(t)$ and t are k-compatible.

Let q be a quasistate with $\Diamond_k C \in t$ for some $t \in q$. A quasistate q' is a *witness* for the triple $(\Diamond_k C, t, q)$ *iff* q and q' are k-compatible and there is $t' \in q'$ such that t and t' are k-compatible and $C \in t'$.

The algorithm starts with the set M_0 of all quasistates for $(C, \mathcal{T})$ and generates a sequence of sets $M_0 \supseteq M_1 \supseteq M_2 \ldots$. In each step the set M_{j+1} is obtained from M_j by eliminating a quasistate q which violates the following condition. For every $t \in q$ and every $\Diamond_k D \in con(C, \mathcal{T})$:

(EL:) if $\Diamond_k D \in t$ then there is a witness for $(\Diamond_k D, t, q)$ in M_j.

The algorithm stops when $M_{j+1} = M_j$, yielding M_j *the final set* computed by the algorithm, and returns 'C is satisfiable w.r.t. $\mathcal{T}$' iff there exists a quasistate $q \in M_j$ and a type $t \in q$ such that $C \in t$. In the next lemma we demonstrate the correctness of the algorithm:

Lemma 2. *The algorithm returns 'C is satisfiable w.r.t. $\mathcal{T}$'* iff *there exists a $(\mathbf{K}_n)_{\mathcal{ALC}}$-model of $\mathcal{T}$ satisfying C.*

Proof. ($\Rightarrow$) Suppose M_j is the final set computed by the algorithm. We construct a $(\mathbf{K}_n)_{\mathcal{ALC}}$-model $\mathfrak{M} = (\mathfrak{C}, \{R_k\}_{1 \leq k \leq n}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c \in \mathfrak{C}})$ inductively. First define the $\mathbf{K}$-frame $(\mathfrak{C}, \{R_k\}_{1 \leq k \leq n})$ by iteratively creating and relating elements of $\mathfrak{C}$ as follows:

- pick a quasistate $q \in M_j$, such that $C \in t$ for some $t \in q$ and create its copy q^* in $\mathfrak{C}$;
- for every $1 \leq k \leq n$, every $q \in \mathfrak{C}$, if $\Diamond_k C \in t$ for some $t \in q$, then pick a witness for $(\Diamond_k C, t, q)$ from M_j, create its new copy q^* in $\mathfrak{C}$ and set $q R_k q^*$.

A *run* ρ through $\mathfrak{C}$ is a choice function which for every $q \in \mathfrak{C}$ selects a type $\rho(q) \in q$. A set of runs $\mathfrak{R}$ is *coherent iff* the following conditions are satisfied:

- for every $q \in \mathfrak{C}$ and every $t \in q$ there is a run $\rho \in \mathfrak{R}$, such that $\rho(q) = t$,
- for every $\rho \in \mathfrak{R}$, $1 \leq k \leq n$ and $q, q' \in \mathfrak{C}$ such that $q R_k q'$ it holds that $\rho(q)$ and $\rho(q')$ are k-compatible,
- for every $\rho \in \mathfrak{R}$, $\Diamond_k D \in con(C, \mathcal{T})$ and $q \in \mathfrak{C}$, if $\Diamond_k D \in \rho(q)$ then there exists $q' \in \mathfrak{C}$ such that $q R_k q'$ and $D \in \rho(q')$.

We let $\Delta = \mathfrak{R}$, for a coherent set of runs $\mathfrak{R}$ through $\mathfrak{C}$, and associate with every $q \in \mathfrak{C}$ an interpretation function:

- for every concept name $A \in con(C, \mathcal{T})$:
  $A^{\mathcal{I}(q)} = \{\rho \in \mathfrak{R} \mid A \in \rho(q)\}$;
- for every role name $r \in rol(C, \mathcal{T})$:
  for every $\rho \in \mathfrak{R}$ and $\exists r.D \in \rho(q)$ pick $\rho' \in \mathfrak{R}$ such that $\{D\} \cup \{\neg C \mid \neg \exists r.C \in \rho(q)\} \subseteq \rho'(q)$ and set $\langle \rho, \rho' \rangle \in r^{\mathcal{I}(q)}$.

By structural induction on $(C, \mathcal{T})$ it follows that $\mathfrak{M}$ is indeed a model for $\mathcal{T}$ satisfying C.

($\Leftarrow$) Let $\mathfrak{M} = (\mathfrak{C}, \{R_k\}_{1 \leq k \leq n}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c \in \mathfrak{C}})$ be a $(\mathbf{K}_n)_{\mathcal{ALC}}$-model of $\mathcal{T}$ satisfying C. We show that there is a subset M of the initial set of all quasistates M_0 such that none of its elements can be eliminated by the algorithm and for some

$q \in M$ there is a $t \in q$ such that $C \in t$, so that the algorithm has to return 'C is satisfiable w.r.t. $\mathcal{T}$'.

Let $\mathbf{t}$ be a function mapping every pair from $\Delta \times \mathfrak{C}$ to the concept type from $\prod(C, \mathcal{T})$ determined by the interpretation $\mathfrak{M}$, i.e., for every $d \in \Delta$ and $c \in \mathfrak{C}$ we pose:

- $C \in \mathbf{t}(d, c)$ iff $d \in C^{\mathcal{I}(c)}$, for every $C \in con(C, \mathcal{T})$,

With every world $c \in \mathfrak{C}$ we can then associate the set of concept types $c_{\mathbf{t}} = \{\mathbf{t}(d, c) \mid d \in \Delta\}$ that are represented by the domain individuals in it. Every such set is thus simply a quasistate. Finally we fix $M = \{c_{\mathbf{t}} \mid c \in \mathfrak{C}\}$, so that M is a collection of quasistates represented in the original model. Clearly one of the elements of M contains a type t, such that $C \in t$, since $\mathfrak{M}$ was satisfying C at the first place. Also, all types in all quasistates contain the TBox concept $C_{\mathcal{T}}$. Moreover, it follows naturally, that the conditions (**QS**) and (**EL**) must be satisfied by all types and all quasistates, hence none of them can be eliminated. ❑

Since the number of types for $(C, \mathcal{T})$ equals $2^{|con(C,\mathcal{T})|}$ and $|con(C, \mathcal{T})| \leq 2\ell(C, \mathcal{T})$, where $\ell(C, \mathcal{T})$ denotes the length (number of symbols) of $(C, \mathcal{T})$, then the number of all quasistates is bounded by $2^{2^{2\ell(C,\mathcal{T})}}$. In the worst case, in order to verify whether the elimination criterion applies to a quasistate at a given stage of the run of the algorithm, it is necessary to compare each of its types against all types from the remaining quasistates, where each comparison can be performed in the polynomial time. Thus the whole algorithm cannot take more than $((2^{2^{2\ell(C,\mathcal{T})}} \cdot 2^{|con(C,\mathcal{T})|}) \cdot 2^{|con(C,\mathcal{T})|}) \cdot 2^{2^{2\ell(C,\mathcal{T})}}$ steps in total to terminate, and thus remains clearly in $O(2^{2^{\ell(C,\mathcal{T})}})$.

Theorem 4. *Deciding concept satisfiability in* $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ *w.r.t. global TBoxes and only with local roles is* 2EXPTIME*-hard.*

The proof is based on reduction of the word problem of an exponentially bounded *Alternating Turing Machine* (ATM), which is known to be 2EXPTIME-hard [21]. One initial observation that will be useful in the reduction is that $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ is Kripke-complete w.r.t. the class of infinite intransitive trees with a constant branching factor, determined by the number of context modalities.

Proposition 4. *A concept* C *is satisfiable w.r.t. a global TBox* $\mathcal{T}$ *in* $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ iff *it is satisfied w.r.t.* $\mathcal{T}$ *in some model* $\mathfrak{M} = (\mathfrak{C}, \{<_i\}_{1 \leq i \leq n}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c \in \mathfrak{C}})$*, such that* $\langle \mathfrak{C}, \bigcup\{<_i\}_{1 \leq i \leq n}\rangle$ *is a tree, every world in* $\mathfrak{C}$ *has exactly one* $<_i$*-successor, for each* $i \in (1, n)$*, and for* $i \neq j$*,* $<_i$*- and* $<_j$*-successors are different.*

Models based on such trees can be easily obtained from the ones we have discussed so far by using the standard unraveling technique. Thus, in what follows, we can safely focus only on $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$*-tree-models.*

Alternating Turing Machines. An ATM is a tuple $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$, where:

- Q is a set of states containing pairwise disjoint sets of *existential states* $Q_\exists$, *universal states* $Q_\forall$, and *halting states* $\{q_a, q_r\}$, where q_a is an *accepting* and q_r a *rejecting* state;
- Σ is an *input alphabet* and Γ a *working alphabet*, containing the *blank symbol* $\varnothing$, such that $\Sigma \subseteq \Gamma$;
- $q_0 \in Q_\exists \cup Q_\forall$ is the *initial state*;
- δ is a *transition relation*, which to every pair $(q, a) \in (Q_\exists \cup Q_\forall) \times \Gamma$, assigns at least one triple $(q', b, m) \in Q \times \Gamma \times \{l, n, r\}$. The triple describes the transition to state q', involving overwriting of symbol a with b and a shift of the head to the left ($m = l$), to the right ($m = r$) or no shift ($m = n$). If q is a halting state then the set of possible transitions $\delta(q, a)$ for every $a \in \Gamma$ is empty.

A *configuration* of an ATM is given as a sequence wqw', where $w, w' \in (\Gamma \setminus \{\varnothing\})^*$ and $q \in Q$, which says that the tape contains the word ww' (possibly followed by blank symbols), the machine is in state q and the head of the machine is on the leftmost symbol of w'. A succeeding configuration is defined by transitions δ, where the head of the machine reads and writes the symbols on the tape. A configuration wqw' is a *halting* one if $q = q_a$ (*accepting configuration*) or if $q = q_r$ (*rejecting configuration*).

Without loss of generality we adopt a somewhat simplified and more convenient setup for ATMs presented in [15]. An *ATM computation tree* of $\mathcal{M}$ is a finite tree whose nodes are labeled with configurations and such that the following conditions are satisfied:

- the root contains the *initial configuration* $q_0 w$, where w is of length n,
- every configuration wqw' on the tree, where ww' is of length at most 2^n, is succeed by:
  - at least one successor configuration, whenever $q \in Q_\exists$,
  - all successor configurations, whenever $q \in Q_\forall$,
- all leaves are labeled with halting configurations.

A tree is *accepting iff* all the leaves are labeled with accepting configurations and *rejecting* otherwise. An ATM *accepts* an input w iff there exists an accepting ATM tree with $q_0 w$ as its initial configuration. The set of all words accepted by an ATM $\mathcal{M}$ is denoted as the language $L(\mathcal{M})$. According to [21, Theorem 3.4], the problem of deciding whether $w \in L(\mathcal{M})$, for w and $\mathcal{M}$ complying to the requirements described above, is 2ExpTime-hard.

Reduction. Technically the reduction is quite involved but its conceptual core is straightforward. We use separate **DAlt** modalities for representing symbols of the alphabet and possible transitions. By isolating specific fragments of $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models we can thus embed the syntactic structure of an ATM computation tree (see Figure 2). At the same time, using special counting concepts, which enable traversing this structure downwards and upwards, we align the succeeding configurations semantically, ensuring they satisfy the constraints of the respective ATM transitions (see Figure 3).

Let $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$ be an ATM and w the word for which we want to decide whether $w \in L(\mathcal{M})$. In the following we will construct a TBox $\mathcal{T}_\mathcal{M}$ and a concept $C_{\mathcal{M},w}$, of a total polynomial size in the size of the input, such that $w \in L(\mathcal{M})$ *iff* $C_{\mathcal{M},w}$ is satisfiable w.r.t. global $\mathcal{T}_\mathcal{M}$ in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$. The encoding is constructed incrementally and provided with extensive explanations on the way.

First we define the set of **DAlt** modal operators:

alphabet modalities: $\bigcirc_a$, for every $a \in \Gamma$,
transition modalities: $\bigcirc_{q,a,m}$, for every $(q, a, m) \in \Theta$, where $\Theta = \{(q, a, m) \mid (q', b, q, a, m) \in \delta$ for any $b \in \Gamma$ and $q' \in Q\}$,

and introduce the following abbreviations (for any concept B):

$$\Box B = \bigsqcap_{a\in\Gamma} \bigcirc_a B,$$
$$\Diamond B = \bigsqcup_{a\in\Gamma} \bigcirc_a B,$$
$$\blacksquare B = \bigsqcap_{(q,a,m)\in\Theta} \bigcirc_{q,a,m} B,$$
$$\blacklozenge B = \bigsqcup_{(q,a,m)\in\Theta} \bigcirc_{q,a,m} B.$$

In the encoding we use several counters, consisting of a number of inclusions of a total polynomial size, which allow to identify distances on the branches of the same fixed length 2^n. Constraints (5)-(9) implement an exemplary *downward counter*, based on atomic concepts X_i, for $1 \leq i \leq n$, which simulate bits in a binary number. The counting is initiated on $d \in \Delta$ whenever d instantiates concept $Count_d$. In every successor **DAlt**-world along the alphabet modalities, d becomes then an instance of a concept description, representing the consecutive number, which uniquely determines the distance from the world in which the counting was initiated. The counter turns the full loop, back to $Count_d$, in periods of 2^n.

$$Count_d \equiv \bigsqcap_{j=1}^{n} \neg X_j, \tag{5}$$

$$\neg X_i \sqcap \neg X_j \sqsubseteq \Box\neg X_i, \text{ for every } 1 \leq j < i \leq n, \tag{6}$$

$$X_i \sqcap \neg X_j \sqsubseteq \Box X_i, \text{ for every } 1 \leq j < i \leq n, \tag{7}$$

$$\neg X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \Box X_j, \text{ for every } 1 \leq j \leq n, \tag{8}$$

$$X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \Box\neg X_j, \text{ for every } 1 \leq j \leq n. \tag{9}$$

An alternative *upward counter*, initiated with $Count_u$ and implemented via template (10)-(14), behaves exactly the same way, with the only difference that the counting proceeds along the alphabet modalities *up* the branch of the model.

$$Count_u \equiv \bigsqcap_{j=1}^{n} X_j, \tag{10}$$

$$\Diamond(X_i \sqcap X_j) \sqsubseteq X_i, \text{ for every } 1 \leq j < i \leq n, \tag{11}$$

$$\Diamond(\neg X_i \sqcap X_j) \sqsubseteq \neg X_i, \text{ for every } 1 \leq j < i \leq n, \tag{12}$$

$$\Diamond(X_j \sqcap \neg X_{j-1} \sqcap \ldots \sqcap \neg X_1) \sqsubseteq \neg X_j, \text{ for every } 1 \leq j \leq n, \tag{13}$$

$$\Diamond(\neg X_j \sqcap \neg X_{j-1} \sqcap \ldots \sqcap \neg X_1) \sqsubseteq X_j, \text{ for every } 1 \leq j \leq n. \tag{14}$$

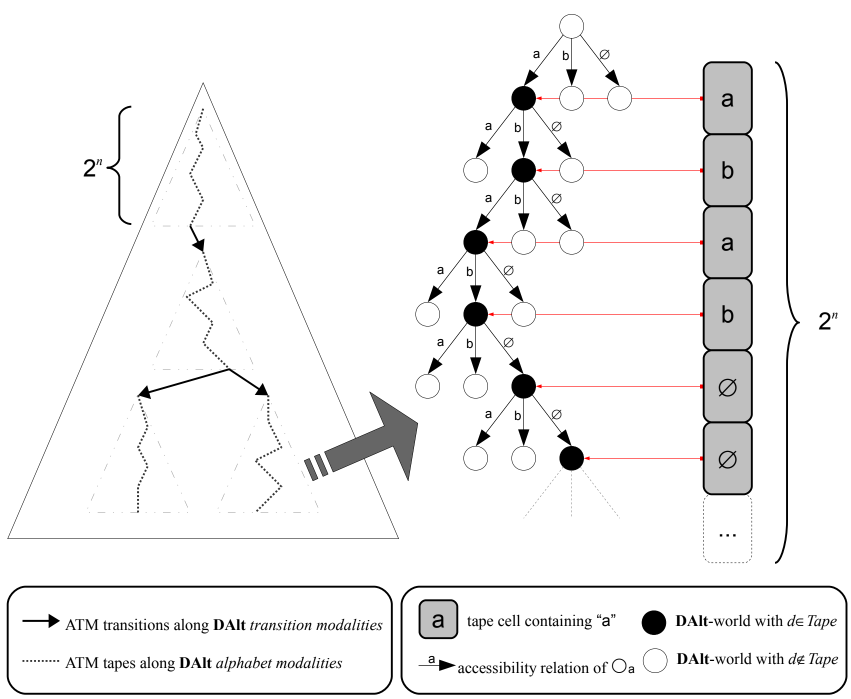

Fig. 2. Embedding of ATM computation trees (left) and ATM tapes (right) in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models.

We can now introduce a fresh downward counter $Count_d^{tape}$:

$$Count_d^{tape} \equiv \bigsqcap_{j=1}^{n} \neg R_j, \tag{15}$$

and define constraints which encode a single tape on a branch of a model. In (16) we define the beginning of such a tape, in (17) its end, while with (18)-(20) we ensure that there is a unique path connecting the two. Note that whenever an individual d instantiates concept *StartTape*, it becomes an instance of *Tape* for exactly 2^n succeeding worlds along a unique path of alphabet modalities. We will consider such a path as determining the content of the tape, as presented in Figure 2. In fact, in our models we will need only one such individual which will single out the whole structure of the ATM tree. Constraint (20) ensures that the blank symbol is followed only by blank symbols on the tape.

$$\mathit{StartTape} \equiv \mathit{Tape} \sqcap \mathit{Count}_d^{tape}, \tag{16}$$

$$\mathit{EndTape} \equiv \mathit{Tape} \sqcap \Diamond \mathit{Count}_d^{tape}, \tag{17}$$

$$\mathit{Tape} \sqcap \neg \mathit{EndTape} \sqsubseteq \Diamond \mathit{Tape}, \tag{18}$$

$$\Diamond(\mathit{Tape} \sqcap \neg \mathit{StartTape}) \sqsubseteq \mathit{Tape}, \tag{19}$$

$$\bigcirc_a \mathit{Tape} \sqcap \bigcirc_b \mathit{Tape} \sqsubseteq \bot, \text{ for every } a \neq b \in \mathit{\Gamma}, \tag{20}$$

$$\bigcirc_\varnothing (\mathit{Tape} \sqcap \bigcirc_a \mathit{Tape}) \sqsubseteq \bot, \text{ for every } a \neq \varnothing \in \mathit{\Gamma}. \tag{21}$$

Further, we implement the transitions by transferring the necessary information downwards or upwards the branches of a $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-model, as depicted in Figure 3.

For the downward part, we introduce new concept names Q_q for every $q \in Q$ and $M_{q,a,m}$ for every $(q, a, m) \in \Theta$, as well as a fresh downward counter Count_d^{head} (22) for measuring the distance from the original position of the head. The Q_q concepts denote the current state and the position of the head, while the others serve for carrying the information about the following transitions. Information about the transitions is generated depending on whether the state is universal (23) or existential (24) and then carried to the end of the tape. There the transitions take place (25)-(26) and new tapes are initiated.

$$\mathit{Count}_d^{head} \equiv \bigsqcap_{j=1}^{n} \neg S_j, \tag{22}$$

$$\bigcirc_a(Q_q \sqcap \mathit{Tape}) \sqsubseteq \bigcirc_a\Big(\bigsqcap_{(q'b'm)\in\delta(q,a)} M_{q',b,m} \sqcap \mathit{Count}_d^{head}\Big), \quad \text{for every } a \in \mathit{\Gamma}, q \in Q_\forall, \tag{23}$$

$$\bigcirc_a(Q_q \sqcap \mathit{Tape}) \sqsubseteq \bigcirc_a\Big(\bigsqcup_{(q'b'm)\in\delta(q,a)} M_{q',b,m} \sqcap \mathit{Count}_d^{head}\Big), \quad \text{for every } a \in \mathit{\Gamma}, q \in Q_\exists, \tag{24}$$

$$M_{q,a,m} \sqsubseteq \Box M_{q,a,m}, \tag{25}$$

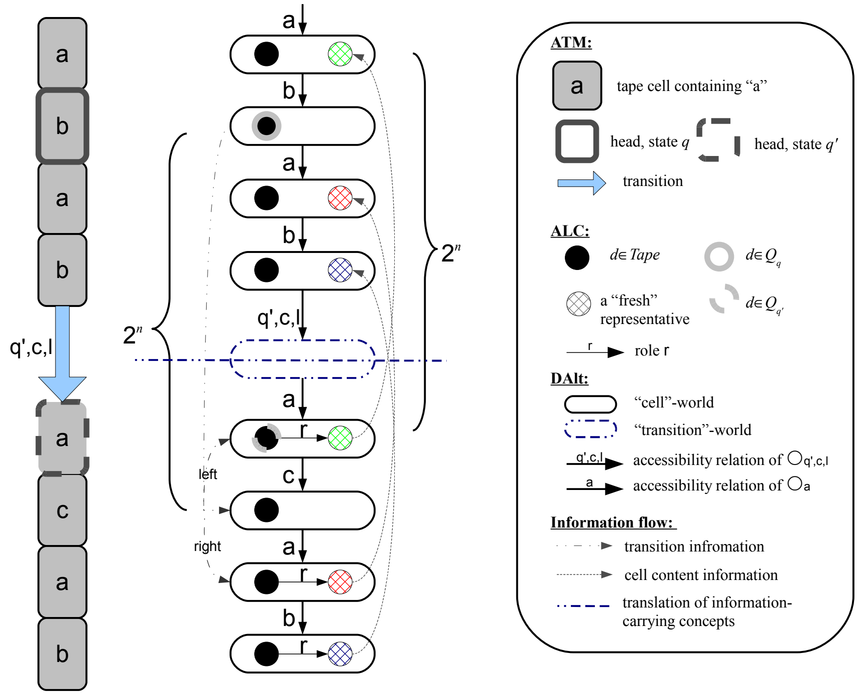

Fig. 3. A transition between succeeding configurations in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models for $n = 2$ and $(q', c, l) \in \delta(q, b)$.

$$M_{q,a,m} \sqcap \mathit{EndTape} \sqsubseteq \bigcirc_{q,a,m} \Diamond \mathit{StartTape}, \text{ for every } (q, a, m) \in \Theta. \tag{26}$$

Note that, once we move along a transition modality, starting a new offspring of the computation, the concepts $M_{q,a,m}$ as well as the counters are not carried along. This is intended, as we want to avoid potential clashes with the information generated on the succeeding tapes. However, we still need to inform the new offsprings about their configurations. To this end we create copies $N_{q,a,m}$ for all concepts $M_{q,a,m}$, which continue to carry their information over the new tape (25)-(26). Further we introduce a fresh downward counter Count_d^{*head}, which proceeds with the counting exactly from the point where the previous head counter terminated (29)-(31). Finally, the constraints (32)-(33) introduce some handy abbreviations which will be used for imposing the new configuration.

$$M_{q,a,m} \sqsubseteq \bigcirc_{q,a,m} N_{q,a,m}, \tag{27}$$

$$N_{q,a,m} \sqsubseteq \Box N_{q,a,m}, \tag{28}$$

$$\mathit{Count}_d^{*head} \equiv \bigsqcap_{j=1}^{n} \neg T_j, \tag{29}$$

$$S_i \sqsubseteq \blacksquare T_i, \text{ for every } 1 \leq i \leq n, \tag{30}$$

$$\neg S_i \sqsubseteq \blacksquare \neg T_i, \text{ for every } 1 \leq i \leq n, \tag{31}$$

$$Count_d^{*head} - 1 \equiv Head^l, \tag{32}$$

$$Count_d^{*head} \equiv Head^n, \tag{33}$$

$$Count_d^{*head} + 1 \equiv Head^r. \tag{34}$$

The necessary changes in the configuration are imposed through constraints (35)-(36), which place the head in the appropriate position, marking it with the new state concept, and force the old position to be overwritten with the new symbol. The inclusions (37)-(38) ensure that the transition does not push the head beyond the tape.

$$N_{q,a,m} \sqcap Tape \sqcap Head^m \sqsubseteq Q_q, \text{ for every } (q, a, m) \in \Theta, \tag{35}$$

$$\bigcirc_b (N_{q,a,m} \sqcap Tape \sqcap Head^n) \sqsubseteq \bot, \text{ for every } (q, a, m) \in \Theta \text{ and } b \neq a \in \Gamma, \tag{36}$$

$$Head^n \sqcap StartTape \sqsubseteq \neg N_{q,a,l}, \text{ for every } q \in Q, a \in \Gamma, \tag{37}$$

$$Head^n \sqcap EndTape \sqsubseteq \neg N_{q,a,r}, \text{ for every } q \in Q, a \in \Gamma. \tag{38}$$

In the opposite direction we will transfer the information about the content of the cells which are not meant to change during the transition. This information is carried by newly generated 'representatives', i.e., new r-successors of the individual instantiating *Tape*. Observe that since our models are tree-shaped, it follows that whenever the representative reaches the 2^n-th ancestor world (upwards the alphabet modalities and one transition modality), it is exactly the world which holds the previous version of the represented cell. This enables us to align the content of the two versions. In a similar way as before, we introduce two fresh upward counters which are synchronized at the point of transition (39)-(42).

$$Count_u^{cell} \equiv \prod_{j=1}^{n} U_j, \tag{39}$$

$$Count_u^{*cell} \equiv \prod_{j=1}^{n} V_j, \tag{40}$$

$$\blacklozenge U_i \sqsubseteq V_i, \text{ for every } 1 \leq i \leq n, \tag{41}$$

$$\blacklozenge \neg U_i \sqsubseteq \neg V_i, \text{ for every } 1 \leq i \leq n. \tag{42}$$

At the same time, for each $a \in \Gamma$ we introduce two concept names W_a, S_a, whose interpretation is propagated upwards the alphabet modalities (43)-(44) and aligned at the transition point (45). Constraint (46) generates a representative of each cell (except for the one that has been changed, marked with the concept $Head^n$), and equips it with the concept W describing the cell's content.

Once this information arrives to the previous version of that cell we prevent the cells from having different content (47).

$$\Diamond W_a \sqsubseteq W_a, \text{ for every } a \in \Gamma, \tag{43}$$

$$\Diamond S_a \sqsubseteq S_a, \text{ for every } a \in \Gamma, \tag{44}$$

$$\blacklozenge W_a \sqsubseteq S_a, \text{ for every } a \in \Gamma, \tag{45}$$

$$\bigcirc_a (\mathit{Tape} \sqcap \neg \mathit{Head}^n) \sqsubseteq \bigcirc_a \exists r.(\mathit{Count}_u^{cell} \sqcap W_a), \text{ for every } a \in \Gamma, \tag{46}$$

$$\bigcirc_a (S_b \sqcap \mathit{Count}_u^{*cell}) \sqsubseteq \bot, \text{ for every } b \neq a \in \Gamma. \tag{47}$$

Finally, it suffices to ensure that nowhere in the model is the rejecting state satisfied.

$$\top \sqsubseteq \neg Q_{q_r} \tag{48}$$

This completes the construction of the TBox $\mathcal{T}_\mathcal{M}$. The initial configuration $q_0 w$ is encoded as concept $C_{\mathcal{M},w}$. Let $w = a_1 \ldots a_n$. For $2 \leq i \leq n$ define recursively:

$$\begin{aligned} A_i &= \bigcirc_{a_i}(\mathit{Tape} \sqcap A_{i+1}) \\ A_{n+1} &= \bigcirc_{\varnothing} \mathit{Tape} \end{aligned}$$

Then $C_{\mathcal{M},w} = \bigcirc_{a_1}(\mathit{StartTape} \sqcap Q_{q_0} \sqcap A_2)$. We conclude by demonstrating validity of the target claim:

Lemma 3. $w \in L(\mathcal{M})$ *iff* $C_{\mathcal{M},w}$ *is satisfiable w.r.t. global* $\mathcal{T}_\mathcal{M}$ *in* $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$

Proof. ($\Rightarrow$) Suppose $w \in L(\mathcal{M})$ and T is an ATM computation tree accepting w. We roughly sketch the construction of a model $\mathfrak{M} = (\mathfrak{C}, \{<_x\}_{x \in \Gamma \cup \Theta}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c \in \mathfrak{C}})$ of $\mathcal{T}_\mathcal{M}$ satisfying $C_{\mathcal{M},w}$.

We assume that each tape associated with a configuration in T is of length exactly 2^n. Let $t(i, wqw')$ be a function returning the i-th symbol from the tape containing ww', and $h(wqw')$ a function returning the position of the head over that tape. Let $q_0 w$ be the initial configuration and $c \in \mathfrak{C}$ the root of $\mathfrak{M}$. Then for some $d \in \Delta$ set $d \in C_{\mathcal{M},w}^{\mathcal{I}(c)}$. Then encode the tape of $q_0 w$ starting from c, according to the following inductive procedure. Given a tape of wqw' and the world $c \in \mathfrak{C}$ in which the encoding starts, set $i := 1$ and $x := c$ and proceed recursively until $i = 2^n + 1$:

1. pick $c \in \mathfrak{C}$ such that $x <_{t(i,wqw')} c$;
2. set $d \in \mathit{Tape}^{\mathcal{I}(c)}$;
3. if $i = 1$ then set $d \in \mathit{StartTape}^{\mathcal{I}(c)}$ and $d \in (\mathit{Count}_d^{tape})^{\mathcal{I}(c)}$;
4. if $i = h(wqw')$ then set $d \in Q_q^{\mathcal{I}(c)}$, $d \in (\mathit{Count}_d^{head})^{\mathcal{I}(c)}$ and for all transitions (q, a, m) from wqw' performed on T, $d \in M_{q,a,m}^{\mathcal{I}(c)}$;
5. if $i = 2^n$ then set $d \in \mathit{EndTape}^{\mathcal{I}(c)}$;
6. set $\langle d, e \rangle \in r^{\mathcal{I}(c)}$ for some fresh $e \in \Delta$, $e \in W_{t(i,wqw')}^{\mathcal{I}(c)}$ and $e \in (\mathit{Count}_u^{cell})^{\mathcal{I}(c)}$
7. set $i := i + 1$ and $x := c$;

Then for every transition (q, a, m) from wqw' in T, resulting in the succeeding configuration $vq'v'$, pick the world $c \in \mathfrak{C}$ such that $x <_{q,a,m} c$ and repeat the procedure above for the tape of $vq'v'$ starting from the world c. Once the halting configurations are encoded, fix the interpretations of the bit concepts associated with the respective counters and propagate the interpretations of selected concepts as follows:

- $M_{q,a,m}$ and $N_{q,a,m}$ for every $(q, a, m) \in \Theta$: downwards along relations $<_x$ for all $x \in \Gamma$;
- W_a and S_a for every $a \in \Gamma$: upwards along relations the $<_x$ for all $x \in \Gamma$;

In the worlds representing the transition points, ensure the proper alignment of the interpretations of the concept pairs $M_{q,a,m}$ – $N_{q,a,m}$ and W_a – S_a, as well as the bit concepts of the counters $Count_d^{head}$ – $Count_d^{*head}$ and $Count_d^{cell}$ – $Count_d^{*cell}$.

($\Leftarrow$) This direction of the claim follows straightforwardly from the reduction. In order to retrieve an ATM tree accepting w from a $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-model we only need to pick an individual d, such that $d \in C_{\mathcal{M},w}^{\mathcal{I}(c_0)}$ and follow the paths of worlds $c \in \mathfrak{C}$ for which $d \in Tape^{\mathcal{I}(c)}$, just as presented in Figure 2. On the way we collect information about the entire configuration. Two important comments are in order. First, note that the reduction is somewhat underconstrained in the sense that the models might represent also some surplus states or transitions. However, the proper computation tree, i.e., the one directly enforced by the encoding, has to appear within this structure. Secondly, we recall that the ATM trees we consider are all finite. Since the transitions in the reduction properly simulate those of an ATM, therefore the trees embedded in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models have to be also finite, even though the models themselves are always infinite. ❑

Theorem 6. *Deciding satisfiability of an $\mathcal{ALC}_{\mathcal{ALC}}$ knowledge base in which object roles are interpreted locally is* 2ExpTime*-complete.*

In the following we prove only the upper bound. Let $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ be an $\mathcal{ALC}_{\mathcal{ALC}}$ knowledge base. We devise a quasimodel elimination algorithm which decides satisfiability of $\mathcal{K}$ in at most double exponential time in the size of $\mathcal{K}$. W.l.o.g. we assume that all occurrences of $[\cdot]_{\boldsymbol{p}}$ in $\mathcal{K}$ are replaced with $\neg\langle\cdot\rangle_{\boldsymbol{p}}\neg$, and all occurrences of $\forall r.$ ($\forall \boldsymbol{r}.$) with $\neg\exists r.\neg$ ($\neg\exists \boldsymbol{r}.\neg$). By $sub_o(\mathcal{K})$ we denote the set of all object (sub)formulas occurring in $\mathcal{K}$, closed under negation, by $con_c(\mathcal{K})$ and $con_o(\mathcal{K})$ the sets of all c-concepts and o-concepts occurring in $\mathcal{K}$, respectively, closed under negation; similarly by $rol_c(\mathcal{K})$ and $rol_o(\mathcal{K})$ the sets of context and object role names, and by $obj_c(\mathcal{K})$ and $obj_o(\mathcal{K})$ the sets of context and object individual names.

A *context type* for $\mathcal{K}$ is a pair $\langle c, f\rangle$, such that $c \subseteq con_c(\mathcal{K})$ and $f \subseteq sub_o(\mathcal{K})$, where:

- $A \in c$ iff $\neg A \notin c$, for all $A \in con_c(\mathcal{K})$,
- $C \sqcap D \in c$ iff $\{C, D\} \subseteq c$, for all $C \sqcap D \in con_c(\mathcal{K})$,

– $\varphi \in f$ iff $\neg\varphi \notin f$, for all $\varphi \in sub_o(\mathcal{K})$,
– $\varphi \wedge \psi \in f$ iff $\{\varphi, \psi\} \subseteq f$, for all $\varphi \wedge \psi \in sub_o(\mathcal{K})$.

An *object type* for $\mathcal{K}$ is a subset $t \subseteq con_o(\mathcal{K})$, where:

– $A \in t$ iff $\neg A \notin t$, for all $A \in con_o(\mathcal{K})$,
– $C \sqcap D \in t$ iff $\{C, D\} \subseteq t$, for all $C \sqcap D \in con_o(\mathcal{K})$.

A *quasistate* for $\mathcal{K}$ is a tuple $q = \langle c_q, f_q, O_q, \pi_q \rangle$, where $\langle c_q, f_q \rangle$ is a context type, O_q is a non-empty set of object types for $\mathcal{K}$, and π_q is a function associating with every name from $obj_o(\mathcal{K})$ an object type from O_q. We say that q is *saturated iff* for every $t \in O_q$:

(qS) if $\exists r.D \in t$ then there is a type $t' \in O_q$ s.t. $\{D\} \cup \{\neg C \mid \neg\exists r.C \in t\} \subseteq t'$.

We call q *coherent iff* the following conditions hold:

(qC1) for every $a : C \in sub_o(\mathcal{K})$, $a : C \in f_q$ *iff* $C \in \pi_q(a)$,
(qC2) for every $r(a, b) \in sub_o(\mathcal{K})$, if $r(a, b) \in f_q$ then $\{\neg C \mid \neg\exists r.C \in \pi_q(a)\} \subseteq \pi_q(b)$,
(qC3) for every $C \sqsubseteq D \in sub_o(\mathcal{K})$, $C \sqsubseteq D \in f_q$ *iff* for every $t \in O_q$, if $C \in t$ then $D \in t$.

Two quasistates $q = \langle c_q, f_q, O_q, \pi_q \rangle$ and $q' = \langle c'_q, f'_q, O'_q, \pi'_q \rangle$ are said to be $\boldsymbol{p}$*-compatible*, for some $\boldsymbol{p} \in rol_c(\mathcal{K})$, *iff*

– $\{\neg \boldsymbol{C} \mid \neg\exists \boldsymbol{p}.\boldsymbol{C} \in c_q\} \subseteq c'_q$,
– $\{\neg\varphi \mid \neg\langle \boldsymbol{C} \rangle_{\boldsymbol{p}} \varphi \in f_q\} \subseteq f'_q$, for all $\boldsymbol{C} \in c'_q$,
– there exists a relation $g \subseteq O_q \times O'_q$, called a $\boldsymbol{p}$*-linkage*, such that:
  - for every $t \in O_q$ there exists a $t' \in O'_q$ such that $\langle t, t' \rangle \in g$,
  - for every $t' \in O'_q$ there exists a $t \in O_q$ such that $\langle t, t' \rangle \in g$,
  - for every $a \in obj_o(\mathcal{K})$ it holds that $\langle \pi_q(a), \pi'_q(a) \rangle \in g$,
  - for every $t \in O_q$ and $t' \in O'_q$ such that $\langle t, t' \rangle \in g$, if $\neg\langle \boldsymbol{C} \rangle_{\boldsymbol{p}} D \in t$ and $\boldsymbol{C} \in c'_q$ then $\neg D \in t'$.

A set of quasistates Q is *saturated iff* for every quasistate $q \in Q$:

(QS1) for every $\exists \boldsymbol{p}.\boldsymbol{C} \in c_q$ there is a quasistate $q' \in Q$ such that $\boldsymbol{C} \in c'_q$ and q is $\boldsymbol{p}$-compatible with q';
(QS2) for every $\langle \boldsymbol{C} \rangle_{\boldsymbol{p}} \varphi \in f_q$ there is a quasistate $q' \in Q$ such that $\boldsymbol{C} \in c'_q$, $\varphi \in f'_q$ and q is $\boldsymbol{p}$-compatible with q';
(QS3) for every $t \in O_q$ and $\langle \boldsymbol{C} \rangle_{\boldsymbol{p}} D \in t$ there is a quasistate $q' \in Q$ such that $\boldsymbol{C} \in c'_q$ and q is $\boldsymbol{p}$-compatible with q' via some $\boldsymbol{p}$-linkage g such that $D \in g(t)$.

A *quasimodel* for $\mathcal{K}$ is a pair $\mathfrak{N} = \langle Q, \gamma \rangle$, such that Q is a non-empty, saturated set of saturated and coherent quasistates for $\mathcal{K}$, γ is a function associating with every name from $obj_c(\mathcal{K})$ a quasistate from Q, and the following conditions are satisfied:

(M1) if $\boldsymbol{a} : \boldsymbol{C} \in \mathcal{C}$ then $\boldsymbol{C} \in c_q$ for $q \in Q$ such that $\gamma(\boldsymbol{a}) = q$,
(M2) if the set $R_{\boldsymbol{a},\boldsymbol{b}} = \{\boldsymbol{p} \mid \boldsymbol{p}(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{C}\}$ is non-empty then $\gamma(\boldsymbol{a})$ is $\boldsymbol{p}$-compatible with $\gamma(\boldsymbol{b})$ for all $\boldsymbol{p} \in R_{\boldsymbol{a},\boldsymbol{b}}$ and there exists a common $\boldsymbol{p}$-linkage between $\gamma(\boldsymbol{a})$ and $\gamma(\boldsymbol{b})$ for all such $\boldsymbol{p}$,
(M3) if $\boldsymbol{C} \sqsubseteq \boldsymbol{D} \in \mathcal{C}$ then for every $q \in Q$, if $\boldsymbol{C} \in c_q$ then $\boldsymbol{D} \in c_q$,
(M4) if $\boldsymbol{a} : \varphi \in \mathcal{O}$ then $\varphi \in f_q$, for $q \in Q$ such that $\gamma(\boldsymbol{a}) = q$,
(M5) if $\boldsymbol{C} : \varphi \in \mathcal{O}$ then $\varphi \in f_q$, for every $q \in Q$ such that $\boldsymbol{C} \in c_q$.

We can now prove the quasimodel lemma.

Lemma 4. *There is a quasimodel for* $\mathcal{K}$ iff *there is an* $\mathcal{ALC}_{\mathcal{ALC}}$*-model of* $\mathcal{K}$.

Proof. An important observation which we exploit in this proof is that the constraints imposed on quasimodels ensure existence of certain specific quasistates which represent role successors in the context dimension. This is the case in conditions (QS1)-(QS3), which enforce existence of a $\boldsymbol{p}$-compatible quasistate for each existential restriction (or diamond operator) over a context role occurring in the knowledge base. Similarly in (M2), which ensures that a pair of quasistates representing named contexts is compatible via the same linkage for all roles relating these contexts. To ease reference to these specific elements we amend the corresponding conditions with the following definitions:

(QS1*) in such case call q' a *witness* for $(\exists \boldsymbol{p}.\boldsymbol{C}, q)$ and a $\boldsymbol{p}$-linkage g, enforced by the condition, a *witnessing linkage*;
(QS2*) in such case call q' a *witness* for $(\langle \boldsymbol{C} \rangle_{\boldsymbol{p}} \varphi, q)$ and a $\boldsymbol{p}$-linkage g, enforced by the condition, a *witnessing linkage*;
(QS3*) in such case call q' a *witness* for $(\langle \boldsymbol{C} \rangle_{\boldsymbol{p}} D, t, q)$ and a $\boldsymbol{p}$-linkage g, enforced by the condition, a *witnessing linkage*;
(M2*) in such case call a common linkage g, enforced by the condition, a *witnessing linkage.*

($\Rightarrow$) Suppose $\mathfrak{N} = \langle Q, \gamma \rangle$ is a quasimodel for $\mathcal{K} = (\mathcal{C}, \mathcal{O})$. We will sketch the construction of an $\mathcal{ALC}_{\mathcal{ALC}}$-model $\mathfrak{M} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c \in \mathfrak{C}})$ of $\mathcal{K}$. We start by constructing an interpretation of the context dimension $(\mathfrak{C}, \cdot^{\mathcal{J}})$. First, for every $\boldsymbol{a} \in obj_c(\mathcal{K})$ add a copy q' of the quasistate $q = \gamma(\boldsymbol{a})$ to $\mathfrak{C}$ and set $\boldsymbol{a}^{\mathcal{J}} = q'$. In case $obj_c(\mathcal{K}) = \emptyset$ set $\mathfrak{C} = \{q\}$ for any $q \in Q$. Then fix an interpretation of roles, while iteratively extending $\mathfrak{C}$. For every $\boldsymbol{p} \in rol_c(\mathcal{K})$:

- for every $\boldsymbol{p}(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{C}$ set $\langle \boldsymbol{a}^{\mathcal{J}}, \boldsymbol{b}^{\mathcal{J}} \rangle \in \boldsymbol{p}^{\mathcal{J}}$;
- for every $q \in \mathfrak{C}$:
  - for every $\exists \boldsymbol{p}.\boldsymbol{D} \in c_q$ pick a witness for $(\exists \boldsymbol{p}.\boldsymbol{D}, q)$ from Q, add its copy q' to $\mathfrak{C}$ and set $\langle q, q' \rangle \in \boldsymbol{p}^{\mathcal{J}}$;
  - for every $\langle \boldsymbol{C} \rangle_{\boldsymbol{p}} \varphi \in f_q$ pick a witness for $(\langle \boldsymbol{C} \rangle_{\boldsymbol{p}} \varphi, q)$ from Q, add its copy q' to $\mathfrak{C}$ and set $\langle q, q' \rangle \in \boldsymbol{p}^{\mathcal{J}}$;
  - for every $t \in O_q$ and $\langle \boldsymbol{C} \rangle_{\boldsymbol{p}} D \in t$ pick a witness for $(\langle \boldsymbol{C} \rangle_{\boldsymbol{p}} D, t, q)$ from Q, add its copy q' to $\mathfrak{C}$ and set $\langle q, q' \rangle \in \boldsymbol{p}^{\mathcal{J}}$.

Finally for every $\boldsymbol{A} \in con_c(\mathcal{K})$ set $\boldsymbol{A}^{\mathcal{J}} = \{q \in \mathfrak{C} \mid \boldsymbol{A} \in c_q\}$. It follows by structural induction that all complex c-concepts are satisfied by $\mathfrak{M}$ in the expected contexts, and therefore, since $\mathfrak{N}$ satisfies conditions (M1)-(M3), all axioms from the context knowledge base $\mathcal{C}$ must be satisfied. Now let us turn to the object dimension.

A *run* ρ through $\mathfrak{C}$ is a choice function which for every $q \in \mathfrak{C}$ selects an object type $\rho(q) \in O_q$. Runs are used for representing the behavior of object individuals across contexts. The easiest way to properly constrain this behavior is by employing the witnessing linkages introduced above. Note that the way the interpretation $(\mathfrak{C}, \cdot^{\mathcal{J}})$ has been constructed ensures that for every two contexts related by some role there exists a witnessing linkage we can refer to in order to align the interpretations of object individuals inhabiting these contexts. A set of runs $\mathfrak{R}$ is *coherent iff* the following conditions are satisfied:

- for every $q \in \mathfrak{C}$ and every $t \in O_q$ there is a run $\rho \in \mathfrak{R}$, such that $\rho(q) = t$,
- for every $a \in obj_o(\mathcal{K})$, there is exactly one run $\rho_a \in \mathfrak{R}$ such that $\rho_a(q) = \pi_q(a)$ for all $q \in \mathfrak{C}$,
- for every $q, q' \in \mathfrak{C}$ such that $\langle q, q' \rangle \in \boldsymbol{p}^{\mathcal{J}}$, for some $\boldsymbol{p} \in rol_c(\mathcal{K})$, relation $\{\langle \rho(q), \rho(q') \rangle \mid \rho \in \mathfrak{R}\}$ coincides with the witnessing linkage between q and q';

We let $\Delta = \mathfrak{R}$, for a coherent set of runs $\mathfrak{R}$ through $\mathfrak{C}$, and associate with every $q \in \mathfrak{C}$ an interpretation function:

- for every individual name $a \in obj_o(\mathcal{K})$ set $a^{\mathcal{I}(q)} = \rho_a(q)$
- for every concept name $A \in con_o(\mathcal{K})$:
  $A^{\mathcal{I}(q)} = \{\rho \in \mathfrak{R} \mid A \in \rho(q)\}$;
- for every role name $r \in rol_o(\mathcal{K})$:
  for every $\rho \in \mathfrak{R}$ and $\exists r.D \in \rho(q)$ pick $\rho' \in \mathfrak{R}$ such that $\{D\} \cup \{\neg C \mid \neg \exists r.C \in \rho(q)\} \subseteq \rho'(q)$ and set $\langle \rho, \rho' \rangle \in r^{\mathcal{I}(q)}$.

Note that by aligning runs with the witnessing linkages we automatically ensure that an object obtains compatible interpretations in every two related contexts. In particular whenever $d \in (\langle \boldsymbol{C} \rangle_{\boldsymbol{p}} D)^{\mathcal{I}(c)}$ for some $d \in \Delta$ and $c \in \mathfrak{C}$, there has to exist a context $c' \in \boldsymbol{C}^{\mathcal{J}}$ accessible from c through $\boldsymbol{p}$ in which $d \in D^{\mathcal{I}(c')}$. By the same token, whenever $d \in ([\boldsymbol{C}]_{\boldsymbol{p}} D)^{\mathcal{I}(c)}$ then $d \in D^{\mathcal{I}(c')}$ in all contexts $c' \in \boldsymbol{C}^{\mathcal{J}}$ accessible through $\boldsymbol{p}$ from c. By structural induction it is not difficult to see that all complex o-concepts are satisfied by $\mathfrak{M}$ as expected — by the designated objects in the designated contexts — and thus, since $\mathfrak{N}$ satisfies conditions (M4)-(M5) and (qC1)-(qC3), all axioms from the object knowledge base $\mathcal{O}$ must be also satisfied.

($\Leftarrow$) This direction is straightforward. Let $\mathfrak{M} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c \in \mathfrak{C}})$ be an $\mathcal{ALC}_{\mathcal{ALC}}$-model of $\mathcal{K}$. We construct a quasimodel $\mathfrak{N} = (Q, \gamma)$ for $\mathcal{K}$ as follows. Let $\mathbf{t}$ be a function mapping every context from $\mathfrak{C}$ to its type determined by the interpretation $\mathfrak{M}$, i.e., for every $c \in \mathfrak{C}$, set $\mathbf{t}(c) = \langle t_c, f_c \rangle$ where t_c and f_c have to satisfy the constraints:

- $\boldsymbol{C} \in t_c$ iff $c \in \boldsymbol{C}^{\mathcal{J}}$, for every $\boldsymbol{C} \in con_c(\mathcal{K})$,

– $\varphi \in f_c$ iff $\mathcal{I}, c \models \varphi$, for every $\varphi \in sub_o(\mathcal{K})$.

In the same way we use $\mathbf{t}$ to denote object types for objects. For every object-context pair $\langle d, c\rangle \in \Delta \times \mathfrak{C}$ we define $\mathbf{t}(d, c)$ as:

– $C \in \mathbf{t}(d, c)$ iff $d \in C^{\mathcal{I}(c)}$, for every $C \in con_o(\mathcal{K})$,

Further, for every $c \in \mathfrak{C}$ let $O_c = \{\mathbf{t}(d, c) \mid d \in \Delta\}$ be the set of object types represented in the context c, and let π_c be a function mapping every $a \in obj_o(\mathcal{K})$ to its type in that context, i.e., $\pi_c(a) = \mathbf{t}(a^{\mathcal{I}(c)}, c)$. We can then define a quasistate for every $c \in \mathfrak{C}$ as $q_c = \langle t_c, f_c, O_c, \pi_c\rangle$, where $\mathbf{t}(c) = \langle t_c, f_c\rangle$. Finally, let $Q = \{q_c \mid c \in \mathfrak{C}\}$ and γ be a function mapping every $\boldsymbol{a} \in obj_c(\mathcal{K})$ to its corresponding quasistate from Q, i.e., $\gamma(\boldsymbol{a}) = q_c$ whenever $\boldsymbol{a}^{\mathcal{J}} = c$. Clearly $\mathfrak{M} = (Q, \gamma)$ is a quasimodel for $\mathcal{K}$. In particular, it is guaranteed that for all existential restrictions (and diamond expressions) occurring in the context and object types from the quasistates, there must exist suitable witnesses and witnessing linkages, and thus that all conditions constituting quasimodels have to be satisfied. ❑

The basic, brute-force algorithm deciding whether a quasimodel for $\mathcal{K}$ exists is again a straightforward extension of the type elimination method, similar to the one used in the proof of Theorem 3. We start by enumerating all quasimodel candidates $\mathfrak{N}_1, \ldots, \mathfrak{N}_N$, such that every candidate $\mathfrak{N}_i = (Q, \gamma)$ consists of the set of all possible quasistates Q and a unique mapping associating quasistates with context names. It is easy to see that $N = |Q|^{|obj_c(\mathcal{K})|}$. Recall that a quasistate is defined as $q = (c_q, f_q, O_q, \pi_q)$, hence the number of all possible quasistates $|Q|$ for $\mathcal{K}$ is:

$$|Q| = (2^{|con_c(\mathcal{K})|} \cdot 2^{|sub_o(\mathcal{K})|} \cdot 2^{2^{|con_o(\mathcal{K})|}})^{|obj_o(\mathcal{K})|}.$$

Observe also that for $\ell(\mathcal{K})$, denoting the length (number of symbols) of $\mathcal{K}$, we have:

$$\begin{gathered} |con_o(\mathcal{K})| + |con_c(\mathcal{K})| \leq 2\ell(\mathcal{K}), \\ |obj_o(\mathcal{K})| + |obj_c(\mathcal{K})| \leq \ell(\mathcal{K}), \\ |sub_o(\mathcal{K})| \leq 2\ell(\mathcal{K}). \end{gathered}$$

The total number of quasimodel candidates is hence bounded by:

$$N \leq (2^{2^{2\ell(\mathcal{K})} \cdot \ell(\mathcal{K}) + 4\ell(\mathcal{K})^2})^{\ell(\mathcal{K})} \leq 2^{2^{2\ell(\mathcal{K})} \cdot \ell(\mathcal{K})^2 + 4\ell(\mathcal{K})^3},$$

and so $N \in O(2^{2^{\ell(\mathcal{K})}})$. Given a candidate the algorithm eliminates it whenever any of the conditions (M1)-(M5) is violated, or iteratively eliminates the quasistates and object types from the quasistates which violate any of the conditions (qS), (qC1)-(qC3), (QC1)-(QC3). It succeeds *iff* the following conditions are met:

– no more object types nor quasistates can be eliminated,
– there is at least one quasistate left and every quasistate contains at least one object type,

- for every $\boldsymbol{a} \in obj_c(\mathcal{K})$ the quasistate $\gamma(\boldsymbol{a})$ is not eliminated,
- for every quasistate q and $a \in obj_o(\mathcal{K})$ the object type $\pi_q(\boldsymbol{a})$ is not eliminated.

In such case the result of elimination is clearly a quasimodel and the search is finished. Else the algorithm processes the consecutive quasimodel candidate, and so on until no more candidates are left. If all candidates are eliminated with no success then there obviously exists no quasimodel. Note, that similarly to the algorithm from Theorem 3, the number of steps necessary for completing elimination in one candidate quasimodel is at most double exponential in $\ell(\mathcal{K})$. Hence the whole algorithm runs also in double exponential time.

Theorem 7. *The language $L_{1\times 2}$ interpreted in product models is exactly as expressive as the concept language of $\mathcal{ALC}_{\mathcal{ALC}}$ interpreted in models with rigid interpretations of object roles.*

Proof. To prove the claim we will show that (1) for every $\mathcal{ALC}_{\mathcal{ALC}}$ concept D there is a concept $C \in L_{1\times 2}$ and, conversely, (2) for every concept $C \in L_{1\times 2}$ there is an $\mathcal{ALC}_{\mathcal{ALC}}$ concept D, such that C is satisfied in some product model *iff* D is satisfied in some $\mathcal{ALC}_{\mathcal{ALC}}$ model in which object roles are interpreted rigidly. Note that we consider C and D to be arbitrary syntactically well-formed concepts. In case of $\mathcal{ALC}_{\mathcal{ALC}}$ this includes both c- and o-concepts.

We start by defining a mapping between two kinds of interpretations w.r.t. the vocabulary in C and D. We say that a product interpretation $\mathcal{P}$ and an $\mathcal{ALC}_{\mathcal{ALC}}$ interpretation $\mathfrak{M}$ in which object roles are interpreted rigidly are *matching iff* $\mathcal{P} = (\mathfrak{C} \times \Delta, \cdot^{\mathcal{P}})$, $\mathfrak{M} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c\in\mathfrak{C}})$, and the functions $\cdot^{\mathcal{P}}$, $\cdot^{\mathcal{J}}$ and $\cdot^{\mathcal{I}}$ are related as follows:

- for every $\boldsymbol{p}$: $\langle v, w\rangle \in \boldsymbol{p}^{\mathcal{J}}$ *iff* $\langle (v,x), (w,y)\rangle \in \boldsymbol{p}^{\mathcal{P}}$, for any $x, y \in \Delta$,
- for every $\boldsymbol{A}$: $v \in \boldsymbol{A}^{\mathcal{J}}$ *iff* $(v,x) \in \boldsymbol{A}^{\mathcal{P}}$, for any $x \in \Delta$,
- for every r and $c \in \mathfrak{C}$: $\langle x, y\rangle \in r^{\mathcal{I}(c)}$ *iff* $\langle (c,x), (c,y)\rangle \in r^{\mathcal{P}}$,
- for every A and $c \in \mathfrak{C}$: $x \in A^{\mathcal{I}(c)}$ *iff* $(c,x) \in A^{\mathcal{P}}$.

Obviously every product interpretation has a matching $\mathcal{ALC}_{\mathcal{ALC}}$ interpretation and vice versa.

Case (1) is straightforward. Let D be an $\mathcal{ALC}_{\mathcal{ALC}}$ concept. Apply the following rules to all subconcepts D' of D:

- if $D' = \langle \boldsymbol{B}\rangle_{\boldsymbol{p}} C$ then replace it with $\exists \boldsymbol{p}.(\boldsymbol{B} \sqcap C)$,
- if $D' = [\boldsymbol{B}]_{\boldsymbol{p}} C$ then replace it with $\forall \boldsymbol{p}.(\neg \boldsymbol{B} \sqcup C)$.

Let C be the result of the transformation. Clearly, C is a well-formed $L_{1\times 2}$. By structural induction on the concepts it is easy to see that if D is satisfied in some $\mathfrak{M}$ then C is satisfied in the matching product interpretation, and if C is satisfied in some $\mathcal{P}$ then D is satisfied in the matching $\mathcal{ALC}_{\mathcal{ALC}}$ interpretation. In particular, if $x \in (\langle \boldsymbol{B}\rangle_{\boldsymbol{p}} C)^{\mathcal{I}(c)}$ then for some c' we have: $c' \in \boldsymbol{B}^{\mathcal{J}}$, $\langle c, c'\rangle \in \boldsymbol{p}^{\mathcal{J}}$ and $x \in C)^{\mathcal{I}(c')}$. This to the matching product model, where $(c', y) \in \boldsymbol{B}^{\mathcal{P}}$,

$\langle (c,y),(c',y)\rangle \in \boldsymbol{p}^{\mathcal{P}}$, for all $y \in \Delta$, and $(c',x) \in C^{\mathcal{P}}$. Similarly in the opposite direction.

Case (2) is a bit more tedious. Basically, we need to first transform an $L_{1\times 2}$ concept into a form in which concepts of M_C occur right after the restrictions on roles M_R. Then we can smoothly translate them into $\mathcal{ALC}_{\mathcal{ALC}}$ following the opposite transformation to the one used in case (1). Let $C \in L_{1\times 2}$. W.l.o.g. we can assume that $C = \exists \boldsymbol{s}.C'$ for some role name $\boldsymbol{s} \in M_R$ and C' in NNF. We say that a concept B is:

1. in Conjunctive Normal Form (CNF) *iff* $B = \bigsqcap_i \bigsqcup_j B_{ij}$,
2. in Disjunctive Normal Form (DNF) *iff* $B = \bigsqcup_i \bigsqcap_j B_{ij}$,
3. in clausal form *iff* $B = \bigsqcup_i \boldsymbol{B}_i \sqcup \bigsqcup_j B_j$,
4. in conjunctive form *iff* $B = \bigsqcap_i \boldsymbol{B}_i \sqcap \bigsqcap_j B_j$,

where every B_{ij} is a role restriction or a literal, i.e., a concept name, its negation, $\bot$ or $\top$; every B_j is a Σ-literal (including $\bot$ and $\top$), or a role restriction (on any role from $N_R \cup M_R$); every $\boldsymbol{B}_i$ is a Γ-literal (excluding $\bot$ and $\top$). First, we perform a number of equivalence preserving transformations on C. We follow the structure of nestings of role restrictions, starting from the innermost restrictions and proceeding inside-out. On the way we exhaustively apply the π rule:

1. for $\exists r.B$:
   (a) if B is in conjunctive form and $B = \bigsqcap_i \boldsymbol{B}_i \sqcap \bigsqcap_j B_j$, then: $\pi(\exists r.B) = \bigsqcap_i \boldsymbol{B}_i \sqcap \exists r.\bigsqcap_j B_j$,
   (b) if B is in DNF and $B = \bigsqcup_i \bigsqcap_j B_{ij}$ then: $\pi(\exists r.B) = \bigsqcup_i \exists r.\bigsqcap_j B_{ij}$,
   (c) else transform B to DNF and repeat.
2. for $\forall r.B$:
   (a) if B is in clausal form and $B = \bigsqcup_i \boldsymbol{B}_i \sqcup \bigsqcup_j B_j$, then: $\pi(\forall r.B) = \bigsqcup_i \boldsymbol{B}_i \sqcup \exists r.\bigsqcup_j B_j$,
   (b) if B is in CNF and $B = \bigsqcap_i \bigsqcup_j B_{ij}$ then: $\pi(\forall r.B) = \bigsqcap_i \forall r.\bigsqcup_j B_{ij}$,
   (c) else transform B to CNF and repeat.
3. for $\exists \boldsymbol{p}.B$:
   (a) if B is in DNF and $B = \bigsqcup_i \bigsqcap_j B_{ij}$ then: $\pi(\exists \boldsymbol{p}.B) = \bigsqcup_i \exists \boldsymbol{p}.\bigsqcap_j B_{ij}$,
   (b) else transform B to DNF and repeat.
4. for $\forall \boldsymbol{p}.B$:
   (a) if B is in CNF and $B = \bigsqcap_i \bigsqcup_j B_{ij}$ then: $\pi(\forall \boldsymbol{p}.B) = \bigsqcap_i \forall \boldsymbol{p}.\bigsqcup_j B_{ij}$,
   (b) else transform B to CNF and repeat.

Note, that given the difference in the semantics of concept names from N_C and M_C, steps 1a and 2a also preserve equivalence. As a result we obtain a concept in which all concept names of M_C occur only on the first depth inside restrictions on roles M_R. Moreover, in all concepts of the form $\exists \boldsymbol{p}.B$, B is in conjunctive form and in all concepts of the form $\forall \boldsymbol{p}.B$, B is in clausal form. Using these observations we can translate the outcome of the transformation into the syntax of $\mathcal{ALC}_{\mathcal{ALC}}$ by applying the following rules:

1. for $\exists \boldsymbol{p}.B$ where $B = \bigsqcap_i \boldsymbol{B}_i \sqcap \bigsqcap_j B_j$:

(a) if $i \neq 0$ and $j \neq 0$ then $\pi(\exists \boldsymbol{p}.B) = \langle \bigsqcap_i \boldsymbol{B}_i \rangle_{\boldsymbol{p}} \bigsqcap_j B_j$,
(b) if $i = 0$ then $\pi(\exists \boldsymbol{p}.B) = \langle \top \rangle_{\boldsymbol{p}} \bigsqcap_j B_j$,
(c) if $j = 0$ then $\pi(\exists \boldsymbol{p}.B) = \langle \bigsqcap_i \boldsymbol{B}_i \rangle_{\boldsymbol{p}} \top$.

2. for $\forall \boldsymbol{p}.B$ where $B = \bigsqcup_i \boldsymbol{B}_i \sqcap \bigsqcup_j B_j$:
(a) if $i \neq 0$ and $j \neq 0$ then $\pi(\forall \boldsymbol{p}.B) = [\neg(\bigsqcup_i \boldsymbol{B}_i)]_{\boldsymbol{p}} \bigsqcup_j B_j$,
(b) if $i = 0$ then $\pi(\forall \boldsymbol{p}.B) = [\top]_{\boldsymbol{p}} \bigsqcup_j B_j$,
(c) if $j = 0$ then $\pi(\exists \boldsymbol{p}.B) = [\neg(\bigsqcup_i \boldsymbol{B}_i)]_{\boldsymbol{p}} \bot$.

Let D be the result of the translation. Clearly, D is an $\mathcal{ALC}_{\mathcal{ALC}}$ concept. Again it is not difficult to find out by structural induction on the concepts that if D is satisfied in some $\mathfrak{M}$ then C is satisfied in the matching product interpretation, and if C is satisfied in some $\mathcal{P}$ then D is satisfied in the matching $\mathcal{ALC}_{\mathcal{ALC}}$ interpretation. ❑