Description Logics of Context

Szymon Klarman
VU University Amsterdam, The Netherlands
Centre for Artificial Intelligence Research,
CSIR Meraka and UKZN, South Africa
szymon.klarman@gmail.com

Víctor Gutiérrez-Basulto
University of Bremen, Germany
victor@informatik.uni-bremen.de

Abstract

We introduce *Description Logics of Context* (DLCs) — an extension of Description Logics (DLs) for context-based reasoning. Our approach descends from J. McCarthy's tradition of treating contexts as formal objects over which one can quantify and express first-order properties. DLCs are founded in two-dimensional possible world semantics, where one dimension represents a usual object domain and the other a domain of contexts, and accommodate two interacting DL languages — the object and the context language — interpreted over their respective domains. Effectively, DLCs comprise a family of *two-sorted*, *two-dimensional* combinations of pairs of DLs. We argue that this setup ensures a well-grounded, generic framework for capturing and studying mechanisms of contextualization in the DL paradigm. As the main technical contribution, we prove 2ExpTime-completeness of the satisfiability problem in the maximally expressive DLC, based on the DL $\mathcal{SHIO}$. As an interesting corollary, we show that under certain conditions this result holds also for a range of two-dimensional DLs, including the prominent $(\mathbf{K}_n)_{\mathcal{ALC}}$.

1 Introduction

Description Logics (DLs) are popular knowledge representation formalisms, applied successfully in a number of fields as logic tools for designing and operating ontologies, i.e., formal models of terminologies and instance data, representative of particular domains of interest [5]. The Semantic Web is one of the outstanding environments where such ontologies, expressed in the DL-based Web Ontology Languages (OWL), play a key architectural role: facilitating publication of knowledge on the Web in a machine-understandable way [6]. Through the close ties to OWL, DLs effectively provide the Semantic Web with its mathematical foundations and determine the methodology of knowledge modeling and the reasoning regime observed by the ontology-based Web applications. Alongside the benefits gained from this relationship, come also significant limitations inherent to the DL paradigm. One such shortcoming, which we focus on here, is the lack of a generic mechanism for dealing with contextual aspects of knowledge.

Under the standard Kripkean semantics, a DL ontology imposes a unique, global and uniform view on the represented domain. Put technically, the axioms of an ontology are interpreted as unconditionally and universally true in all models of that ontology, e.g., $\mathit{Heart} \sqsubseteq \mathit{HumanOrgan} \in \mathcal{O}$ enforces all domain individuals of type *Heart* to be of type *HumanOrgan* in all possible models of $\mathcal{O}$. Such a representation philosophy is well-suited as long as everyone shares the same conceptual perspective on the domain or if there is no purpose for considering alternative viewpoints. Although these two conditions have traditionally warranted the very use of ontologies in AI and many ontology engineers still chose to abide by them, it is also commonly observed that ever more often they simply fail to be satisfied in practice. Most of times a domain should be modeled differently depending on the context in which it is considered, where the context might depend on a spatio-temporal coordinate, the thematic focus, a subjective perspective of the modeler, the adopted level of granularity of the representation, an intended application of the ontology, etc. For instance, the axiom $\mathit{Heart} \sqsubseteq \mathit{HumanOrgan}$, valid in the domain of human anatomy, should likely lose its generality once a broader perspective of mammal anatomy is considered. Moreover, the intrinsic inability of accounting for contexts in DLs seems to hinder the usability of DLs in two fundamental application scenarios. 1) In principle, it is impossible to create ontologies that would be at the same time broad enough as to capture all relevant information about the domain and yet sufficiently detailed as to cover all context-specific peculiarities in the formal representation of that knowledge. This challenge is commonly faced by the creators of huge knowledge bases, aiming at maximum coverage of the representation, such as SNOMED [40] or Cyc [31], and can easily lead to the development of application-driven mechanisms of contextualization. 2) The second problem concerns the reuse of knowledge from multiple existing sources — such as the numerous DL-based ontologies already published on the Web — in new applications. Naturally, portions of such knowledge retrieved from different ontologies are likely to pertain to different, heterogenous contexts, which are implicitly assumed during the creation of the sources. Consequently, a faithful reuse of such data cannot be achieved without special semantic mechanisms which acknowledge and respect its local, context-specific character [23, 7].

Interestingly, variants of these two problems are well-recognized in the field of knowledge representation and serve as a motivation for two popular theories of context as used in knowledge representation systems. Bouquet *et al.* denote these theories or views as *divide-and-conquer* and *compose-and-conquer* [10] and describe them as follows:

> "[...] According to the first view, which we call *divide-and-conquer*, there is something like a global theory of the world. This global

> theory has an internal structure, and this structure is articulated into a collection of contexts. According to the second view, which we call *compose-and-conquer*, there is not such a thing as a global theory of the world, but only many local theories. Each local theory represents a viewpoint on the world. Also, there may exist relations between local theories that allow a reasoner to (partially) compose them into a more comprehensive view. [...]"

As argued by the authors of the quoted passage, these two theories of context are relevant for problems of two different types and hence they naturally give rise to two diverse sorts of formal solutions. The ongoing research efforts on incorporating contexts into the DL framework clearly exhibit this dichotomy. On the one side, we witness attempts of extending the DL languages with means of modeling the contextual dependence of knowledge on certain implicit states inherent to the semantics, such as levels of abstraction over an ontology [20, 21] or states of some fixed modal dimension — most typically a temporal one [33, 3, 4]. On the other side, there have been several formalisms proposed for supporting the task of context-sensitive integration of classical DL ontologies, including Distributed DLs [9], Package-based DLs [8] or $\mathcal{E}$-connections [30], with no real consensus on the most natural or generic approach.

As both types of solutions are notoriously specialized in their scope, the problem of formulating a broad and well-grounded theory of contexts within the DL paradigm remains open. In this paper, we systematically develop a framework of two-dimensional *Description Logics of Context* (DLCs), which aims at filling this gap, and arguably, bridges the two theories of context under one unifying formal approach. Our proposal is inspired by J. McCarthy's theory of formalizing contexts [34], whose gist is to replace logical formulas φ, as the basic knowledge carriers, with assertions of the form $ist(\boldsymbol{c}, \varphi)$. Such assertions state that φ *is true* in $\boldsymbol{c}$, where $\boldsymbol{c}$ denotes an abstract first-order entity called a *context*. Further, contexts can be on their own described in a first-order language. For instance, the formula:

$$ist(\boldsymbol{c}, Heart(a)) \wedge \boldsymbol{HumanAnatomy}(\boldsymbol{c})$$

states that the object a is a heart in a certain context $\boldsymbol{c}$ of type human anatomy. Formally, we interpret McCarthy's theory in terms of two-dimensional possible world semantics, characteristic of two-dimensional DLs. In DLCs, one semantic dimension represents a usual object domain, while the other a (possibly infinite) domain of contexts. Thus, the notion of context is identified with that of Kripkean *possible world*, which provides the former with a philosophically neutral, yet technically substantial reading, presupposed at the core of McCarthy's theory. Unlike conventional two-dimensional DLs, the DLCs are equipped with two interacting DL languages — the object and the context language — interpreted over their respective domains. These languages allow for explicit modeling of both: the (contextualized) object-level knowledge and the meta-level knowledge, i.e. descriptions of contexts as first-class citizens. Consequently, we define a whole family of *two-sorted*, *two-dimensional* combinations of pairs of DLs, comprising the DLC framework.

Contributions. We propose a novel, generic framework of DLCs for modeling and studying mechanisms of contextualization in the DL paradigm. The framework is derived from two roots: *conceptually* — from McCarthy's theory of formalizing contexts, grounding our approach in a longstanding research tradition in AI; *formally* — from two-dimensional DLs, ensuring strong and well-understood mathematical foundations. We demonstrate the applicability of DLCs to a range of representation problems dealing with contexts. We prove 2ExpTime-completeness of the satisfiability problem in the maximally expressive fragment of the framework studied here, with the object and the context language based on the DL $\mathcal{SHIO}$ [24]. As a corollary, we show that the same result holds also for several underlying two-dimensional DLs with global TBoxes and local interpretation of roles, including the prominent $(\mathbf{K}_n)_{\mathcal{ALC}}$.

Research questions. This paper addresses and provides new answers to the following research questions:

1. How to extend DLs to support the representation of inherently contextualized knowledge?
2. How to use knowledge from coexisting classical DL ontologies while respecting its context-specific scope?
3. Is it possible to capture these two perspectives on contextualization within one unifying formal framework?

Contents. This paper is based on our work reported in the conference publications [27] and [28]. The material is organized as follows. In Section 2 we discuss the formal motivation and the design choices for the proposed framework. In particular, we carefully advocate our interpretation of McCarthy's theory in terms of two-dimensional DL semantics. Section 3 contains the preliminaries for DLs. In Section 4 we present the syntax and semantics for DLCs. Further, in Section 5, we outline possible application scenarios, emphasizing their relation to the *divide-and-conquer* and *compose-and-conquer* theories of context. In Section 6 we address the relationships of DLCs to the well-known two-dimensional DLs $(\mathbf{K}_n)_{\mathcal{L}}$ and $\mathbf{S5}_{\mathcal{L}}$, and elaborate on the computational properties of our logics. Our central result of decidability and the implied upper bound on the satisfiability problem covers the *union* of all DLCs introduced in the conference publications, with the context and object language further extended to the DL $\mathcal{SHIO}$. Finally, in Section 7 we discuss the related work. The paper is concluded in Section 8. Some technical proofs are included in the appendix.

2 Overview and formal motivation

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 [34], 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 [15, 14, 13, 35, 22, 36]. 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. A context is anything that can be denoted by a first-order term and used meaningfully in a statement of the form $ist(\boldsymbol{c}, \varphi)$, saying that formula φ *is true* (*ist*) in context $\boldsymbol{c}$, e.g., $ist(Hamlet,$ 'Hamlet is a prince.') [34, 35, 22, 15]. 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 have properties and can be described. As first-order objects, contexts can be in a natural way described in a first-order language [13, 22]. This allows for addressing them generically through quantified formulas such as $\forall x(\boldsymbol{C}(x) \rightarrow ist(x, \varphi))$, expressing that φ is true in every context of type $\boldsymbol{C}$, e.g., $\forall x(barbershop(x) \rightarrow ist(x,$ 'Main service is a haircut.')).

3. Contexts are organized in relational structures. In the commonsense reasoning, contextual assumptions are dynamically and directionally altered [36, 15]. Contexts are entered and then exited, accessed from other contexts or transcended to broader ones. A simple way of handling their complex organization in formal systems is therefore by means of relational structures, which naturally support representation of diverse relationships and dynamic aspects in first-order domains. On the syntactic level, the use of such structures can be further reflected by permitting nested formulas of type $ist(\boldsymbol{c}, ist(\boldsymbol{d}, \varphi))$. For instance, $ist(France, ist(capital,$ 'The city river is Seine.')) implies that there exists certain relationship between *France* and *capital* such that 'The city river is Seine' is true in the latter context if accessed from (or seen from) the former, but not necessarily when accessed from any other arbitrary context.

The logics proposed in this paper originate as an approach to implementing these postulates within the framework of DLs. As is usually the case, the extent of such implementation can be purposely limited in its scope in order to obtain formalisms of different expressiveness and, consequently, of different computational properties. Thus, instead of a single logic, we derive a family of formalisms generically referred to as *Description Logics of Context* (DLCs). In the following paragraphs we motivate the central design choices we made, starting from basic considerations on the semantics of contextualized knowledge and further tracing their impact on the selection of specific logical languages.

The first, key step to importing McCarthy's theory into the DL framework is to faithfully reinterpret his three postulates on the model-theoretic grounds of DLs. Effectively, such interpretation must lead to a commitment to a specific sort of semantic structures, which need to be taken into account in order to express and interpret contextualized knowledge adequately. Figure 1 illustrates one such structure, based on our intuitive reading of the postulates. It is a formal model of an application domain supporting multiple contexts of representation. As a sample instantiation of this structure, one might consider here a formal

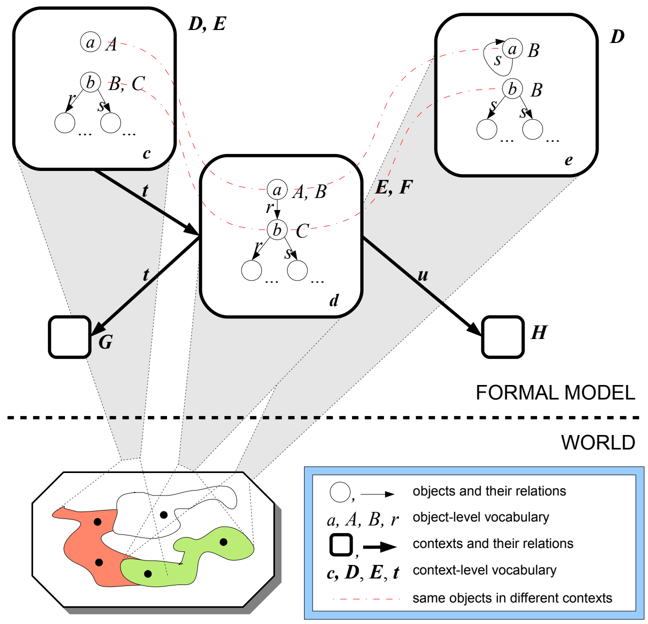

Figure 1: A formal domain model complying to McCarthy's postulates.

description of a society of interconnected agents, each one sustaining its own viewpoint and focus on the represented world.

The model has two apparent levels. The *context-level* consists of context entities (postulate 1), which are possibly interlinked with certain relations (postulate 3) and described in a language containing individual names, concepts and relation names (postulate 2). For instance, context $\boldsymbol{c}$ is of type $\boldsymbol{D}$ and is related to $\boldsymbol{d}$ through a relation of type $\boldsymbol{t}$. Intuitively, each context in the model can be seen as a box carrying a piece of the *object-level* representation. Instead of a unique, global model of the object domain, we associate then a single local model with every context. These models might obviously differ from each other as each of them reflects a specific viewpoint on the object domain. Moreover, they might not necessarily cover the same fragment or aspect of the application domain and not necessarily use the same fragment of the object language for describing it. For instance, objects a and b occur at the same time in contexts $\boldsymbol{c}$, $\boldsymbol{d}$, $\boldsymbol{e}$, but in each of them they are described differently and remain in different relations to other objects.

The central insight emerging under such a perspective is that the semantic structures comprising the model theory of a reasonably expressive DL of context are inherently *two-dimensional*, with one dimension consisting of *domain objects* and the second one of *contexts*.

Once the main characteristics of the intended semantic structures are identified, the next step is to find convenient languages for speaking about them and constraining their possible properties. By the assumption taken in this work,

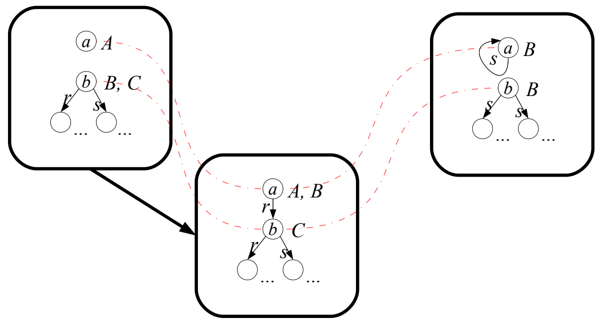

Figure 2: A Kripke model of a two-dimensional DL.

DLs are suitable formalisms for representing the object-level knowledge. The key challenge is then to extend them with additional syntactic means that would facilitate accommodating the context-level information. A first crucial observation in this direction is that the context-level structures, as pictured above, can be seen as Kripke frames, with possible worlds representing context entities and accessibility relations capturing relations between contexts. It is known that such frames can be combined in a product-like fashion with the standard DL interpretations, giving rise to a two-dimensional semantics for DLs with extra modal operators [43, 29] (see Figure 2), which can be used for modeling the dynamics of the object knowledge across the states of the second dimension, for instance time points, as in temporal DLs [33, 4]. As context-dependency bares many apparent similarities to other dynamic aspects of knowledge, building DLCs on top of the standard architecture of two-dimensional DLs seems indeed very natural. Besides the potential benefits of easing the transfer of known results and proof techniques, such setup offers most of all a very clear-cut formal reading of the notoriously elusive notion of *context*. Namely:

CONTEXT = POSSIBLE WORLD

This interpretation of contexts resonates very well with the philosophically neutral and application-agnostic notion of *context-as-formal-object* lying at the heart of McCarthy's theory. At the same time it is technically non-trivial, as it immediately encourages the use of the rich machinery of modal logics for capturing and studying different aspects of contextualization. In particular, various contextualization and lifting operations, i.e. context-sensitive transfers of knowledge between different contexts [34], can be naturally modeled by means of modal operators $\Diamond, \Box$.

Although this approach, based on two-dimensional DLs, is in general very attractive, the serious caveat is that it does not offer a direct methodology for describing contexts *per se*. More precisely, one can easily augment a DL language with modal '*contextualization*' operators for traversing the context dimension of the models and quantifying over the context entities, but it is not possible to explicitly assert properties of the accessed contexts, for instance to express global

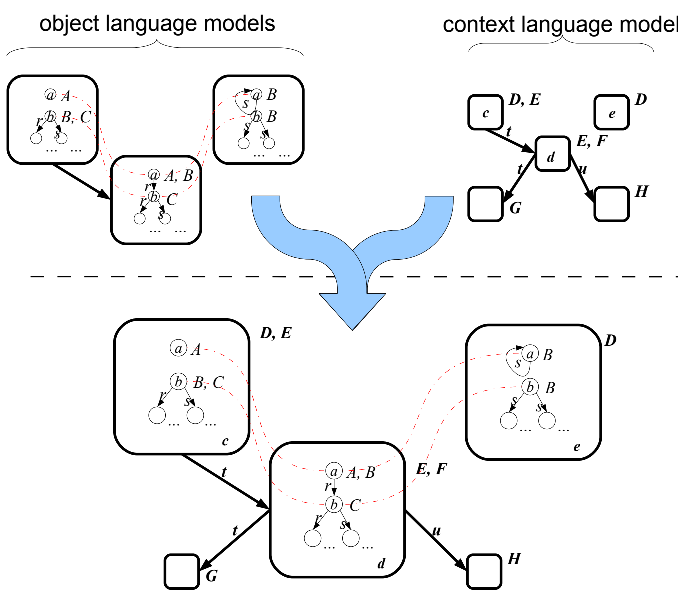

Figure 3: Combining models of two DLs.

contextual dependencies, such as 'In every context of type human anatomy, it holds that: *Heart* $\sqsubseteq$ *HumanOrgan*'. Intuitively, such functionality seems essential for obtaining a fine-grained contextualization machinery. The solution which we propose here is to employ a second DL language for describing the context dimension. As a consequence, we obtain a *two-sorted*, *two-dimensional* logic, where each sort of the language is interpreted over the respective dimension in the semantics. The two languages are suitably integrated on the syntactic and semantic level, so that their models can be eventually combined as presented in Figure 3. The style of combination is fully compatible with the underlying two-dimensional DLs described above. In principle, the two-dimensional models of the object language are embedded in the models of the context language, where possible worlds are mapped on (context) individuals and accessibility relations are mapped on (context) roles. Thus, we are able to show that, depending on the choice of the context operators, our logics are proper extensions of the well-known two-dimensional logics $(\mathbf{K}_n)_{\mathcal{L}}$ or $\mathbf{S5}_{\mathcal{L}}$ [43].

3 Preliminaries

A DL language $\mathcal{L}$ is given by a vocabulary and a set of constructors for composing complex expressions in the language [5]. A vocabulary $\Sigma = (N_C, N_R, N_I)$ consists of a set of *concept names* N_C, a set of *role names* N_R, and a set of *individual names* N_I. The semantics of $\mathcal{L}$ is given through *interpretations* of the form $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, where Δ is a non-empty *domain* of individuals and $\cdot^{\mathcal{I}}$

	Syntax	Semantics
		Concept and role constructors:
(1)	$\top$	Δ
(2)	$C \sqcap D$	$\{x \in \Delta \mid x \in C^{\mathcal{I}} \cap D^{\mathcal{I}}\}$
(3)	$\exists r.C$	$\{x \in \Delta \mid \exists y : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
(4)	$\neg C$	$\{x \in \Delta \mid x \notin C^{\mathcal{I}}\}$
(5)	$\{a\}$	$\{a^{\mathcal{I}}\}$
(6)	r^-	$\{(x,y) \in \Delta \times \Delta \mid (y,x) \in r^{\mathcal{I}}\}$
		Axioms:
(7)	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
(8)	$r(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$
(9)	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
(10)	$r \sqsubseteq s$	$r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$

Table 1: DL constructors and axioms and their semantics.

is an *interpretation function* mapping each $a \in N_I$ to an element $a^{\mathcal{I}} \in \Delta$, each $A \in N_C$ to a subset $A^{\mathcal{I}} \subseteq \Delta$ and each $r \in N_R$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta \times \Delta$. The interpretation function is then inductively extended over complex expressions according to the fixed semantics of the constructors, as presented in Table 1. Different sets of constructors constitute DLs of possibly different expressive power and computational properties. The basic DL $\mathcal{ALC}$ comprises constructors (1)-(4), where one writes $\bot$ as an abbreviation for $\neg\top$, $C \sqcup D$ for $\neg(\neg C \sqcap \neg D)$, and $\forall r.C$ for $\neg\exists r.\neg C$ [38]. The DL $\mathcal{SHIO}$, which is a significant subset of the OWL 2 DL [25], extends $\mathcal{ALC}$ with nominals (5) and role inverses (6). Moreover, it permits the use of a set of transitive role names $N_{R+} \subseteq N_R$, such that for every $r \in N_{R+}$, $r^{\mathcal{I}}$ is a transitive relation.

A *knowledge base* (*ontology*) $\mathcal{K}$ is a finite set of DL axioms, as presented in Table 1. The language $\mathcal{ALC}$ supports axioms (7)-(9), where (7)-(8) are traditionally known as *ABox* axioms, while (9) as *general concept inclusions*, which form a so-called *TBox*. The DL $\mathcal{SHIO}$ enables additionally construction of simple role hierarchies, based on axioms of type (10). An interpretation $\mathcal{I}$ is a *model* of $\mathcal{K}$ whenever it satisfies all its axioms, according to the satisfaction conditions in Table 1. The computational complexity of the central problem of *knowledge base satisfiability*, i.e., deciding whether a given knowledge base has a model, is ExpTime-complete for both $\mathcal{ALC}$ [5] and $\mathcal{SHIO}$ [24].

Two-dimensional DLs are close relatives of product-like combinations of modal logics [29]. They arise from extending the standard DL languages with modal operators applicable to different types of DL expressions, enabling explicit modeling of a variety of intensional aspects of knowledge, e.g., temporal [33], dynamic [44], evolutionary [4], probabilistic [32] and others. Here we are interested in languages with modal operators applied only to concepts.

Definition 1 (Two-dimensional DL) *Let $\mathcal{L}$ be a DL language and $\Diamond_i$, $\Box_i$, for $i \in (1,n)$, be a set of n pairs of modalities of a modal logic L. Then a two-dimensional DL concept language $L_{\mathcal{L}}$ is the smallest set of concepts closed under constructors of $\mathcal{L}$ and two concept constructors:*

$$\Diamond_i C \quad | \quad \Box_i C$$

for any concept $C \in L_{\mathcal{L}}$.

An interpretation of $L_{\mathcal{L}}$ is a tuple $(\mathfrak{W}, \{R_i\}_{i\in(1,n)}, \Delta, \{\mathcal{I}(w)\}_{w\in\mathfrak{W}})$, for some $n \in \mathbb{N}$, where:

- *$\mathfrak{W}$ is a non-empty set of possible worlds,*
- *R_i is an accessibility relation over $\mathfrak{W}$ associated with the operators $\Diamond_i, \Box_i$,*
- *Δ is a non-empty domain of individuals,*
- *for every $w \in \mathfrak{W}$, $\mathcal{I}(w) = (\Delta^{\mathcal{I}(w)}, \cdot^{\mathcal{I}(w)})$ is a DL interpretation in the world w, such that:*
  - $\Delta^{\mathcal{I}(w)} \subseteq \Delta$,
  - $(\Diamond_i C)^{\mathcal{I}(w)} = \{x \in \Delta \mid \exists v : wR_iv \wedge x \in C^{\mathcal{I}(v)}\}$,
  - $(\Box_i C)^{\mathcal{I}(w)} = \{x \in \Delta \mid \forall v : wR_iv \rightarrow x \in C^{\mathcal{I}(v)}\}$.

A two-dimensional DL $L_{\mathcal{L}}$ is interpreted under the constant domain *assumption* iff *the class of interpretations is restricted exactly to those satisfying the condition $\Delta^{\mathcal{I}(w)} = \Delta$, for every $w \in \mathfrak{W}$. Otherwise it is interpreted under the* varying domain *assumption. A DL atom $\alpha \in \Sigma$ is interpreted* rigidly iff *the class of interpretations is restricted exactly to those in which $\alpha^{\mathcal{I}(w)} = \alpha^{\mathcal{I}(v)}$, for every $w, v \in \mathfrak{W}$. Otherwise α is interpreted* locally.

The logic $(\mathbf{K}_n)_{\mathcal{L}}$ is defined as an extension of a DL language $\mathcal{L}$ with n pairs of $\mathbf{K}$-modalities, i.e., operators associated with arbitrary relations $R_i \subseteq \mathfrak{W} \times \mathfrak{W}$. Analogically, the logic $\mathbf{S5}_{\mathcal{L}}$ augments $\mathcal{L}$ with a single pair of $\mathbf{S5}$-modalities associated with the equivalence relation over $\mathfrak{W}$. In the case of all two-dimensional DLs, the definition of the semantics permits certain degrees of freedom regarding domain assumptions (constant vs. varying) and the style of interpretation of atoms across the possible worlds (rigid vs. local). The choice of domain assumptions and the interpretation of individual names are usually only of practical importance and in most cases do not affect the computational properties of the logics [29, 42]. The interpretation of roles, on the other hand, has typically more significant consequences. Rigid roles, turning the formalisms into proper modal products, very often render them also undecidable, and therefore are usually avoided [29]. The basic decision problem for two-dimensional DLs is *concept satisfiability w.r.t. global TBoxes*, i.e., the problem of deciding whether given a concept C and a TBox $\mathcal{T}$ in $L_{\mathcal{L}}$ there exists an interpretation of $L_{\mathcal{L}}$ such that every axiom in $\mathcal{T}$ is satisfied in every possible world in the interpretation and there exists at least one world $w \in \mathfrak{W}$ and an individual $x \in \Delta$ such that $x \in C^{\mathcal{I}(w)}$.

4 Syntax and semantics

A *Description Logic of Context* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ consists of the DL context language $\mathcal{L}_C$, supporting context descriptions, and of the object language $\mathcal{L}_O$ equipped with context operators for representing object knowledge relative to contexts.

Definition 2 ($\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-context language) *The* context language *of* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ *is a DL language* $\mathcal{L}_C$ *over the vocabulary* $\Gamma = (M_C, M_R, M_I)$*, where* M_C *is a set of* context concepts*,* M_R *a set of* context roles *and* M_I *a set of* context names*.*

The object language extends standard DLs with special contextualization operators applicable to concepts.

Definition 3 ($\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-object language) *Let* $\mathcal{L}_O$ *be a DL language over the vocabulary* $\Sigma = (N_C, N_R, N_I)$*. The* object language *of* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ *is the smallest language containing* $\mathcal{L}_O$ *and closed under the constructors of* $\mathcal{L}_O$ *and at least one of the two types —* $\mathfrak{F}_1$ resp. $\mathfrak{F}_2$ *— of concept-forming operators:*

$$\langle \boldsymbol{r}.\boldsymbol{C} \rangle\, D \;\mid\; [\boldsymbol{r}.\boldsymbol{C}]\, D \tag{$\mathfrak{F}_1$}$$

$$\langle \boldsymbol{C} \rangle D \;\mid\; [\boldsymbol{C}] D \tag{$\mathfrak{F}_2$}$$

where $\boldsymbol{C}$ *and* $\boldsymbol{r}$ *are a concept and a role of the context language and* D *is a concept of the object language.*

Intuitively, the concept $\langle \boldsymbol{r}.\boldsymbol{C} \rangle D$ denotes all objects which are D in *some* context of type $\boldsymbol{C}$ accessible from the current one through $\boldsymbol{r}$. Analogically, $[\boldsymbol{r}.\boldsymbol{C}]D$ denotes all objects which are D in *every* such context. In the case of $\mathfrak{F}_2$ operators, the concept $\langle \boldsymbol{C} \rangle D$ denotes all objects which are D in *some* context of type $\boldsymbol{C}$, whereas $[\boldsymbol{C}]D$ all objects which are D in *every* such context. For example, $\langle \boldsymbol{neighbor}.\boldsymbol{Country} \rangle$*Citizen*, refers to the concept *Citizen* in some context of type ***Country*** accessible through the ***neighbor*** relation from the current context. Analogically, $\langle \boldsymbol{HumanAnatomy} \rangle$*Heart* refers to the concept *Heart* in some context of ***HumanAnatomy***.

The operators $\mathfrak{F}_1$ and $\mathfrak{F}_2$ behave similar to the usual modalities of $\mathbf{K}_n$ and **S5**, respectively. In particular, for any $\boldsymbol{r}$ and $\boldsymbol{C}$ the expected dualities hold: $\langle \boldsymbol{r}.\boldsymbol{C} \rangle = \neg[\boldsymbol{r}.\boldsymbol{C}]\neg$ and $\langle \boldsymbol{C} \rangle = \neg[\boldsymbol{C}]\neg$. The only difference is that the contexts (possible worlds) accessed by means of $\mathfrak{F}_1$ or $\mathfrak{F}_2$ are further qualified with concepts of the context language. We formally clarify this relationship in Section 6.

Definition 4 ($\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-knowledge base) *A* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-knowledge base (CKB) *is a pair* $\mathcal{K} = (\mathcal{C}, \mathcal{O})$*, where* $\mathcal{C}$ *is a set of axioms over the context language (in the syntax allowed by* $\mathcal{L}_C$*), and* $\mathcal{O}$ *is a set of formulas of the form:*

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

where φ *is an axiom over the object language (in the syntax allowed by* $\mathcal{L}_O$*),* $\boldsymbol{c} \in M_I$ *and* $\boldsymbol{C}$ *is a concept of the context language.*

A formula $\boldsymbol{c} : \varphi$ states that axiom φ holds in the context denoted by name $\boldsymbol{c}$. Note, that this corresponds directly to McCarthy's $ist(\boldsymbol{c}, \varphi)$. Axioms of the form $\boldsymbol{C} : \varphi$ assert the truth of φ in all contexts of type $\boldsymbol{C}$. For example, the formula $\boldsymbol{Country} : \langle \boldsymbol{neighbor}.\boldsymbol{Country}\rangle Citizen \sqsubseteq NoVisaRequirement$ states that in every country, the citizens of its neighbor countries do not require visas.

The semantics is given through $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-*interpretations* and $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-*models*, which combine the interpretations of $\mathcal{L}_C$ with those of $\mathcal{L}_O$.

Definition 5 ($\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-interpretations) *A $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-interpretation is a tuple $\mathfrak{M} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\mathfrak{C}})$, such that:*

1. *$(\mathfrak{C}, \cdot^{\mathcal{J}})$ is a DL interpretation of $\mathcal{L}_C$, where $\mathfrak{C}$ is a non-empty domain of contexts*[1] *and $\cdot^{\mathcal{J}}$ an interpretation function defined as usual,*

2. *Δ is a non-empty domain of individuals,*

3. *$(\Delta, \cdot^{\mathcal{I}(i)})$, for every $i \in \mathfrak{C}$, is an interpretation of $\mathcal{L}_O$, where $\cdot^{\mathcal{I}(i)}$ is an interpretation function, such that:*

   ($\mathfrak{F}_1$) *for every $\langle \boldsymbol{r}.\boldsymbol{C}\rangle D$ and $[\boldsymbol{r}.\boldsymbol{C}]D$:*
   - $(\langle \boldsymbol{r}.\boldsymbol{C}\rangle\, D)^{\mathcal{I}(i)} = \{x \in \Delta \mid \exists j \in \mathfrak{C} : (i,j) \in \boldsymbol{r}^{\mathcal{J}} \wedge j \in \boldsymbol{C}^{\mathcal{J}} \wedge x \in D^{\mathcal{I}(j)}\}$,
   - $([\boldsymbol{r}.\boldsymbol{C}]\, D)^{\mathcal{I}(i)} = \{x \in \Delta \mid \forall j \in \mathfrak{C} : (i,j) \in \boldsymbol{r}^{\mathcal{J}} \wedge j \in \boldsymbol{C}^{\mathcal{J}} \rightarrow x \in D^{\mathcal{I}(j)}\}$.

   ($\mathfrak{F}_2$) *for every $\langle \boldsymbol{C}\rangle D$ and $[\boldsymbol{C}]D$:*
   - $(\langle \boldsymbol{C}\rangle D)^{\mathcal{I}(i)} = \{x \in \Delta \mid \exists j \in \mathfrak{C} : j \in \boldsymbol{C}^{\mathcal{J}} \wedge x \in D^{\mathcal{I}(j)}\}$,
   - $([\boldsymbol{C}]D)^{\mathcal{I}(i)} = \{x \in \Delta \mid \forall j \in \mathfrak{C} : j \in \boldsymbol{C}^{\mathcal{J}} \rightarrow x \in D^{\mathcal{I}(j)}\}$.

An atom $\alpha \in \Sigma$ of $\mathcal{L}_O$ is interpreted rigidly iff *the class of $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-interpretations is restricted exactly to those in which $\alpha^{\mathcal{I}(i)} = \alpha^{\mathcal{I}(j)}$, for every $i, j \in \mathfrak{C}$. Otherwise α is interpreted* locally.

$\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-interpretations can be seen naturally as extensions of two-dimensional DL interpretations, introduced in Definition 1, where the context domain corresponds to the set of possible worlds. As a consequence, the choice of suitable domain assumptions and the style of interpretation of atoms applies in the same sense also here. The constant domain assumption, adopted by default in the definition above, serves purely technical purposes. Though often unnatural in practical scenarios, it grants greater generality to the complexity results and can be easily relaxed to the varying domain case [29].

The difference between the context operators of type $\mathfrak{F}_1$ and $\mathfrak{F}_2$ lies in the choice of the relational structures observed when quantifying over the context domain. $\mathfrak{F}_1$-operators bind contexts only along the roles of the context language

[1] In some cases it might be useful to distinguish between proper contexts in $\mathfrak{C}$ and other context-level individuals serving merely for context descriptions. For instance, in provenance applications, a context $\boldsymbol{c}$ associated with a knowledge source might be described with axiom $\boldsymbol{hasAuthor}(\boldsymbol{c}, \boldsymbol{henry})$, where $\boldsymbol{henry}$ is an individual related to $\boldsymbol{c}$, but not a context itself [7]. See also the context-level individuals described with G, H in Figure 3.

(as **K**-modalities), while $\mathfrak{F}_2$-operators follow the equivalence universal relation over $\mathfrak{C}$ (as **S5**-modalities). This leads to some clear consequences in the scope and the character of the distribution of the object knowledge over contexts in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-models. For instance, in Figure 1, the concept $\langle \boldsymbol{t}.\boldsymbol{F}\rangle B$ is satisfied by object a only in context $\boldsymbol{c}$, while $\langle \boldsymbol{F}\rangle B$ is satisfied by a in all contexts in the model. From the perspective of McCarthy's theory, employing operators $\mathfrak{F}_2$, rather than $\mathfrak{F}_1$, implies a restriction on postulate (3), permitting only context structures based on equivalence relations, i.e., structures in which every two contexts become in principle accessible to each other. The focus on **K**-like and **S5**-like modalities is driven here mostly by the formal simplicity of the two types of operators and easiness of their integration with the DL semantics. In principle, however, nothing prevents from constructing logics containing contextualization operators which mimic other common modalities.

Finally, we define the notion of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model.

Definition 6 ($\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-models) *A $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-interpretation $\mathfrak{M} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\mathfrak{C}})$ is a* model *of a CKB $\mathcal{K} = (\mathcal{C}, \mathcal{O})$* iff*:*

- *for every $\varphi \in \mathcal{C}$, $(\mathfrak{C}, \cdot^{\mathcal{J}})$ satisfies φ,*
- *for every $\boldsymbol{c} : \varphi \in \mathcal{O}$, $(\Delta, \cdot^{\mathcal{I}(\boldsymbol{c}^{\mathcal{J}})})$ satisfies φ,*
- *for every $\boldsymbol{C} : \varphi \in \mathcal{O}$ and $i \in \mathfrak{C}$, if $i \in \boldsymbol{C}^{\mathcal{J}}$ then $(\Delta, \cdot^{\mathcal{I}(i)})$ satisfies φ.*

Analogically to the standard DLs, we say that a $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-knowledge base $\mathcal{K}$ is *satisfiable iff* $\mathcal{K}$ has a $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model. Likewise, the central reasoning problem in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ is deciding knowledge base satisfiability.

5 Application scenarios

The most common uses of contexts in knowledge systems, as argued in [10], can be classified into two categories, reflecting two prototypical knowledge representation scenarios: divide-and-conquer and compose-and-conquer (see Section 1). The first one concerns the problem of representing inherently contextualized knowledge. The latter, the problem of integrating multiple, non-contextualized knowledge models in a context-sensitive manner. In what follows, we make these two scenarios more concrete by grounding them in the practice of knowledge engineering, we explain how they translate into the DL setting, and outline how they can be supported using DLCs.

5.1 Divide-and-conquer

Picture a complex application domain and a modeler intending to formally represent knowledge about this domain in a possibly generic, application-agnostic manner. His task is to construct a representation model that can be reused

$\mathcal{C}$:	$\boldsymbol{Country}(\boldsymbol{germany})$	(1)
	$\boldsymbol{neighbor}(\boldsymbol{france}, \boldsymbol{germany})$	(2)
$\mathcal{O}$:	$\boldsymbol{germany} : \exists hasParent.Citizen(john)$	(3)
	$\boldsymbol{Country} : \exists hasParent.Citizen \sqsubseteq Citizen$	(4)
	$\boldsymbol{france} : \langle \boldsymbol{neighbor}.\boldsymbol{Country} \rangle Citizen \sqsubseteq NoVisaRequirement$	(5)

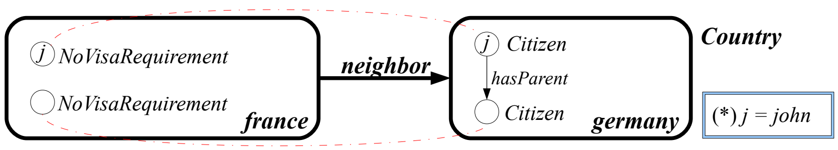

Table 2: A sample CKB in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with $\mathfrak{F}_1$-operators and its possible model.

for different purposes and in different situations, always providing adequate information under the specified conditions. The divide-and-conquer philosophy builds on the observation that in most such cases knowledge is likely to be inherently contextualized, i.e., implicitly partitioned over a collection of interrelated contextual states, which must be taken into account when reasoning about the domain, as they determine which information applies in the given situation. The challenge for the modeler is then to elicit this underlying context structure and explicitly represent it in the model, so that the context-dependency of knowledge is faithfully reflected and operationalized in the system. From the perspective of the DL paradigm, such scenarios require expressive extensions of the standard DL languages, capable of representing contexts. Below we present two examples of applying DLCs to divide-and-conquer scenarios.

A contextualized knowledge base with $\mathfrak{F}_1$-operators. Consider a task of representing knowledge about the visa requirements contextualized with respect to geographic information. This typical instance of the divide-and-conquer scenario invites construction of a complex knowledge model with an inherent contextual layer, which allows for a meaningful qualification of the object knowledge. In Table 2, we define such CKB $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ with $\mathfrak{F}_1$-operators, consisting of the context (geographic) ontology $\mathcal{C}$ and the object (people) ontology $\mathcal{O}$. Visibly, $\boldsymbol{france}$ and $\boldsymbol{germany}$ play here the role of contexts, described in the context language by axioms (1) and (2). In the context of $\boldsymbol{germany}$, it is known that *john* has a parent who is a citizen (3). Since in every $\boldsymbol{Country}$ context — thus including $\boldsymbol{germany}$ — the concept $\exists hasParent.Citizen$ is subsumed by *Citizen* (4), therefore it must be true that *john* is an instance of *Citizen* in $\boldsymbol{germany}$. Finally, since $\boldsymbol{germany}$ is related to $\boldsymbol{france}$ via the role $\boldsymbol{neighbor}$, it follows that *john* (assuming rigid interpretation of this name across contexts) has to be an instance of *NoVisaRequirement* in the context of $\boldsymbol{france}$ (5). A sample $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model of $\mathcal{K}$ is depicted at the bottom of Table 2.

$\mathcal{C}:$	$\boldsymbol{Geometry} \sqsubseteq \boldsymbol{Math}$	(1)
$\mathcal{O}:$	$\boldsymbol{disambiguation} : \mathit{Ring} \sqsubseteq \langle \boldsymbol{Math} \rangle \mathit{Ring} \sqcup \langle \boldsymbol{Astronomy} \rangle \mathit{Ring}$	(2)
	$\boldsymbol{Math} : \mathit{Ring} \sqsubseteq \mathit{AlgebStruct} \sqcup \langle \boldsymbol{Geometry} \rangle \mathit{Annulus}$	(3)
	$\boldsymbol{Astronomy} : \mathit{Ring} \equiv \mathit{PlanetRing}$	(4)

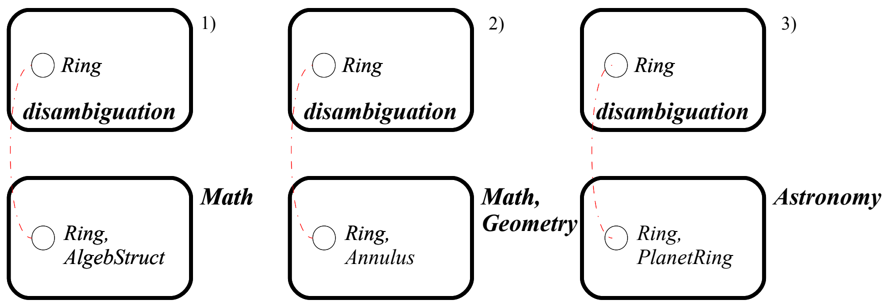

Table 3: A sample CKB in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with $\mathfrak{F}_2$-operators and its possible models.

A contextualized knowledge base with $\mathfrak{F}_2$-operators. In Table 3, we model a piece of information presented on the disambiguation website of Wikipedia on querying for the term *Ring*. In particular, *Ring* is contextualized according to whether it is defined as a mathematical or as an astronomical object.[2] Observe, that the named context ***disambiguation*** provides basic distinction on *Ring* in some ***Math*** context and in some ***Astronomy*** context (2). This is further enhanced, by the distinction defined on the level of all ***Math*** contexts. There, *Ring* denotes either *AlgebStruct* or *Annulus* in some further ***Geometry*** context (3), where ***Geometry*** contexts are known to be a subset of ***Math*** contexts. In case of ***Astronomy*** context, *Ring* is actually equivalent to the concept *PlanetRing* (4). Three possible $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-models of this representation are depicted at the bottom of Table 3. Similarly to the previous example, the divide-and-conquer philosophy is manifested here in the way all context-driven anomalies of the domain knowledge are handled in a monolithic knowledge model.

5.2 Compose-and-conquer

Contrary to what is assumed in the divide-and-conquer scenario, we might observe that many existing knowledge models actually adopt unique, purpose-driven viewpoints on the domain, determined by the particular applications at hand. In certain situations, one might need to reuse a number of such models in one system. To this end, the models must be composed into a reasonably coordinated, single representation. According to the compose-and-conquer philosophy this can be achieved by acknowledging the presence of the contexts implicitly assumed during the creation of each individual model and reflecting on how these

[2] See `http://en.wikipedia.org/wiki/Ring`.

$\mathcal{O}_{\boldsymbol{c}}$:	$\mathit{Staff} \sqsubseteq \exists \mathit{isEmployed}.\mathit{Company}$	(1)
	$\mathit{Staff}(J.\mathit{Smith})$	(2)
$\mathcal{O}_{\boldsymbol{d}}$:	$\mathit{Employee} \sqsubseteq \exists \mathit{employedIn}.\top$	(3)
	$\mathit{Employee}(\mathit{JohnSmith})$	(4)
$\mathcal{O}_{\boldsymbol{e}}$:	$\top : \langle\{\boldsymbol{c}\}\rangle \mathit{Staff} \equiv \langle\{\boldsymbol{d}\}\rangle \mathit{Employee}$	(5)
	$\top : \langle\{\boldsymbol{c}\}\rangle\{J.\mathit{Smith}\} \equiv \langle\{\boldsymbol{d}\}\rangle\{\mathit{JohnSmith}\}$	(6)

Table 4: Integration of ontologies $\mathcal{O}_{\boldsymbol{c}}$, $\mathcal{O}_{\boldsymbol{d}}$ via DLC formulas in $\mathcal{O}_{\boldsymbol{e}}$.

contexts interrelate. The contextualization process is thus considered here as an a posteriori effort of integrating context-specific knowledge models. In the DL paradigm, this problem corresponds to a variety of tasks involving ontology alignment (coordination). Arguably, DLCs can naturally support such scenarios. Observe, that a collection of DL ontologies $\mathcal{O}_1, \ldots, \mathcal{O}_n$ in some language $\mathcal{L}_{\mathcal{O}}$ can be seen as a set of formulas $\mathcal{O} = \{\boldsymbol{c}_i : \varphi \mid \varphi \in \mathcal{O}_i,\ i \in (1, n)\}$ in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, where every ontology is associated with a unique context name. Then using DLC formulas one can impose a number of interesting interoperability constraints over the contents of these ontologies, as presented in the following examples.

Simple vocabulary mappings. A prototypical case of the compose-and-conquer scenario is integration of ontologies by means of external vocabulary mappings. Consider two ontologies $\mathcal{O}_{\boldsymbol{c}}$ and $\mathcal{O}_{\boldsymbol{d}}$ describing overlapping domains, as shown in Table 4. Using context operators $\langle\{\boldsymbol{c}\}\rangle, \langle\{\boldsymbol{d}\}\rangle$ we can align the vocabularies of the two ontologies via simple mappings such as (5)-(6). Given the semantics of DLCs, it follows that *Staff* must have the same meaning in the context $\boldsymbol{c}$ as *Employee* in $\boldsymbol{d}$ (5). Similarly, the denotation of individual names $J.\mathit{Smith}$ and *JohnSmith* is the same across $\boldsymbol{c}$ and $\boldsymbol{d}$ (6). Note, that the context language is restricted here to context names only. In this form, the DLCs provide similar functionality to other known logic-based ontology integration formalisms, which we further discuss in Section 7.

Interoperability constraints for ontology alignment and reuse. The compose-and-conquer philosophy can be also realized using much richer constraints than simple one-to-one mappings used in the previous example. Consider an architecture such as the NCBO BioPortal project[3], which gathers diverse biohealth ontologies, and categorizes them via thematic tags, e.g.: ***Cell***, ***Health***, ***Anatomy***, etc., organized in a meta-ontology. The intention of the project is to facilitate the reuse of the collected resources in new applications. Note, that the division between the context and the object language is already present in the architecture of the BioPortal, this is naturally reflected in the example of Table 5 where (2) maps the concept *Heart* from any ***HumanAnatomy*** ontology to the concept *HumanHeart* in every ***Anatomy*** ontology; (3) imposes the axiom $\mathit{Heart} \sqsubseteq \mathit{Organ}$ of an upper anatomy ontology over all ***Anatomy***

[3]See `http://bioportal.bioontology.org/`.

$\mathcal{C}$:	$\boldsymbol{HumanAnatomy} \sqsubseteq \boldsymbol{Anatomy}$	(1)
$\mathcal{O}$:	$\top : \langle \boldsymbol{HumanAnatomy} \rangle Heart \sqsubseteq [\boldsymbol{Anatomy}] HumanHeart$	(2)
	$\boldsymbol{Anatomy} : Heart \sqsubseteq Organ$	(3)

Table 5: A set of interoperability constraints expressed as a CKB in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ with $\mathfrak{F}_2$-operators.

ontologies, which due to axiom (1) carries over to all ***HumanAnatomy*** ontologies.

In general, $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ provides logic-based explications of some interesting notions, relevant to the problem of semantic interoperability of ontologies, such as:

concept alignment: $\top : \langle \boldsymbol{A} \rangle C \sqsubseteq [\boldsymbol{B}] D$
every instance of C in any ontology of type $\boldsymbol{A}$ is D in every ontology of type $\boldsymbol{B}$

semantic importing: $\boldsymbol{c} : \langle \boldsymbol{A} \rangle C \sqsubseteq D$
every instance of C in any ontology of type $\boldsymbol{A}$ is D in ontology $\boldsymbol{c}$

upper ontology axiom: $\boldsymbol{A} : C \sqsubseteq D$
axiom $C \sqsubseteq D$ holds in every ontology of type $\boldsymbol{A}$

Interoperability constraints for ontology evolving. The context operators can be also interpreted as change operators, in the style of DL of change [4], for instance, for representing and studying dynamic aspects of ontology versioning — especially, when evolutionary constraints apply to a whole collection of semantically interoperable ontologies. Some central issues arising in this setup are integrity (constraining the scope of changes allowed due to versioning), evolvability (ability of coordinating the evolution of ontologies) and formal analysis of differences between the versions [26]. In the examples below, we assume that each ontology version is associated with a unique context, each context concept denotes all versions of a particular ontology and ***updatedBy*** denotes the relation of being an immediate updated version.

version-invariant concepts: $\top : \langle \boldsymbol{A} \rangle C \equiv [\boldsymbol{A}] C$
   C is a version-invariant concept within the scope of versions of type $\boldsymbol{A}$

dynamic analysis: $\top : C \sqcap \langle \boldsymbol{updatedBy}.\top \rangle \neg C \sqsubseteq C^\star$
   $C^\star$ retrieves all instances which are C in some version and evolve into $\neg C$ in some immediate updated version

evolvability constraints: $\boldsymbol{A} : C \sqsubseteq \langle \boldsymbol{updatedBy}.\boldsymbol{B} \rangle D$
   in any version of type $\boldsymbol{A}$, every instance of C has to evolve into D in some immediate updated version of type $\boldsymbol{B}$

Again, this example naturally instantiates the compose-and-conquer scenario. Interoperability constraints are intended here as an external layer, imposed a posteriori over standard DL ontologies in order to elicit and properly coordinate their implicit, context-specific scopes.

6 Formal properties

In this section we touch upon two basic formal properties of DLCs: expressiveness and complexity of reasoning. In addressing these issues, we rely heavily on the fact that the DLC framework is grounded in the well-known two-dimensional DLs [43]. Having such properly established mathematical foundations provides us with two kinds of benefits. Firstly, it allows for a rough demarcation of expressive limits of the DLCs, by direct comparisons to related formalisms that have already been investigated in the literature. Secondly, it enables the adoption of some known proof techniques for studying computational properties of the framework. The results which we deliver here are not exhaustive, but nevertheless, they offer a good limiting characterization of the proposed logics. We show that the expressive power of the full DLC framework properly subsumes the expressiveness of the two-dimensional DLs $(\mathbf{K}_n)_\mathcal{L}$ and $\mathbf{S5}_\mathcal{L}$, and that the problem of satisfiability of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-knowledge bases, for $\mathcal{L}_C$ and $\mathcal{L}_O$ up to the DL $\mathcal{SHIO}$, is decidable — in fact, 2ExpTime-complete.

In the following theorem, we show that concept satisfiability w.r.t. global TBoxes in $(\mathbf{K}_n)_\mathcal{L}$ can be immediately restated as the problem of knowledge base satisfiability in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with $\mathfrak{F}_1$-operators.

Theorem 1 **($(\mathbf{K}_n)_\mathcal{L}$ vs. $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$)** *Deciding concept satisfiability w.r.t. a global TBox in $(\mathbf{K}_n)_\mathcal{L}$ is linearly reducible to knowledge base satisfiability in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, for $\mathcal{L}_O = \mathcal{L}$, with the context operators of type $\mathfrak{F}_1$ only.*

Proof. Let $(C, \mathcal{T})$ be a problem instance in $(\mathbf{K}_n)_\mathcal{L}$. Define the corresponding knowledge base $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ as follows. First, set $\mathcal{C} = \emptyset$ and $\mathcal{O} = \{\top : B \sqsubseteq D \mid B \sqsubseteq D \in \mathcal{T}\} \cup \{\top : (\langle \boldsymbol{s}.\top\rangle C)(a)\}$, for a fresh context role $\boldsymbol{s}$ and a fresh individual object name a. Then, with every pair of $\mathbf{K}$-modalities $\Diamond_i, \Box_i$ in $(\mathbf{K}_n)_\mathcal{L}$ associate a distinct context role name $\boldsymbol{r}_i$ and replace every occurrence of $\Diamond_i$ in $\mathcal{O}$ with $\langle \boldsymbol{r}_i.\top\rangle$ and every occurrence of $\Box_i$ with $[\boldsymbol{r}_i.\top]$. We want to show that C is satisfiable w.r.t. $\mathcal{T}$ in $(\mathbf{K}_n)_\mathcal{L}$ *iff* the resulting knowledge base $\mathcal{K}$ is satisfiable in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$. But this follows immediately by observing the direct correspondence between the semantics of both languages, in particular the semantics of the $\mathbf{K}$-modalities and global TBox axioms in $(\mathbf{K}_n)_\mathcal{L}$, and of the corresponding $\mathfrak{F}_1$-operators and formulas $\top : \varphi$ in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$. In particular, $(\Rightarrow)$ if $(\mathfrak{W}, \{R_i\}_{i\in(1,n)}, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w\in\mathfrak{W}})$ is a model of $\mathcal{T}$ with $x \in C^{\mathcal{I}(w)}$, for some $x \in \Delta$ and $w \in \mathfrak{W}$ then $(\mathfrak{W}, \cdot^\mathcal{J}, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w\in\mathfrak{W}})$ is a model of $\mathcal{K}$, provided that $R_i = (\boldsymbol{r}_i)^\mathcal{J}$, for every $i \in (1,n)$, $x = a^{\mathcal{I}(w)}$, and $(y, x) \in s^{\mathcal{I}(w)}$ for an arbitrary $y \in \Delta$. The $(\Leftarrow)$ direction of the claim follows by reversing this argument. ❏

An analogical reduction follows from $\mathbf{S5}_\mathcal{L}$ to $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with $\mathfrak{F}_2$-operators.

Theorem 2 **($\mathbf{S5}_\mathcal{L}$ vs. $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$)** *Deciding concept satisfiability w.r.t. a global TBox in $\mathbf{S5}_\mathcal{L}$ is linearly reducible to knowledge base satisfiability in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, for $\mathcal{L}_O = \mathcal{L}$, with the context operators of type $\mathfrak{F}_2$ only.*

Proof. Let $(C, \mathcal{T})$ be a problem instance in $\mathbf{S5}_{\mathcal{L}}$. Define the knowledge base $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ by setting $\mathcal{C} = \emptyset$ and $\mathcal{O} = \{\top : B \sqsubseteq D \mid B \sqsubseteq D \in \mathcal{T}\} \cup \{\top : (\langle\top\rangle C)(a)\}$, for a fresh individual name a. Then, replace every occurrence of $\Diamond$ in $\mathcal{O}$ with $\langle\top\rangle$ and every occurrence of $\Box$ with $[\top]$. We want to show that C is satisfiable w.r.t. $\mathcal{T}$ in $\mathbf{S5}_{\mathcal{L}}$ *iff* the resulting knowledge base $\mathcal{K}$ is satisfiable in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$. As in the previous case, we simply observe that the semantics of $\mathbf{S5}$-modalities coincides with that of $\mathfrak{F}_2$-operators. Hence, $(\Rightarrow)$ if $(\mathfrak{W}, R, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w\in\mathfrak{W}})$ is a model of $\mathcal{T}$ with $x \in C^{\mathcal{I}(w)}$, for some $x \in \Delta$ and $w \in \mathfrak{W}$, then $(\mathfrak{W}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w\in\mathfrak{W}})$ with an arbitrary $\mathcal{J}$ is a model of $\mathcal{K}$, provided that $x = a^{\mathcal{I}(w)}$. For the $(\Leftarrow)$ direction we use the equivalence relation R over $\mathfrak{W}$ to obtain a corresponding $\mathbf{S5}_{\mathcal{L}}$-model from a given $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-model of $\mathcal{K}$. ❑

Notably, these correspondences hold regardless of whether object roles are interpreted rigidly in both types of logics or only locally, as stated by default in Definitions 1 and 5.

Observe that for the reductions we use only a residual context language. In the former case we merely require the top concept and a set of context role names, while in the latter only the top concept. Clearly, there is also no need for employing axioms of the context language. This suggests that the expressive power of DLCs might be in general even greater and strictly subsume that of the union of $(\mathbf{K}_n)_{\mathcal{L}}$ and $\mathbf{S5}_{\mathcal{L}}$. Indeed, it is not difficult to instantiate this intuition with concrete examples of properties which are expressible in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$, but cannot be captured by any of the underlying two-dimensional languages. For instance, context names enable certain forms of functional modalities, which point at uniquely identifiable possible worlds, as in axioms of type $\boldsymbol{c} : \varphi$, where the constraint φ is placed exactly over the world named $\boldsymbol{c}$. By allowing nominals in the context language one can further exploit this expressive capability, for instance, to impose cardinality constraints over the possible worlds domain:[4]

$$\top \sqsubseteq \{\boldsymbol{c}\} \sqcup \{\boldsymbol{d}\}, \qquad \{\boldsymbol{c}\} \sqcap \{\boldsymbol{d}\} \sqsubseteq \bot,$$

The context language supports also construction of other complex modalities, as e.g., in the concept:

$$\langle \boldsymbol{A} \rangle C \sqcup [\boldsymbol{A} \sqcap \neg \boldsymbol{B}] C,$$

which describes the set of objects which are C in any context of type $\boldsymbol{A}$ or in all contexts of type $\boldsymbol{A}$ and $\neg\boldsymbol{B}$. Obviously neither $(\mathbf{K}_n)_{\mathcal{L}}$ or $\mathbf{S5}_{\mathcal{L}}$, nor any of the standard two-dimensional DLs, allows for expressing such properties, as they require a more fine-grained mechanism of quantifying over possible worlds, offered by the context language in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$. Although currently we do not have a precise characterization result for the expressiveness of $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$, it seems that at least to some extent its behavior can be simulated in two-dimensional DLs extended with *global concepts*, i.e., concepts C such that for every $w \in \mathfrak{W}$ it

[4]It could therefore be argued that context names introduce certain characteristic features of hybrid logic into the context language, in a similar way as individual object names introduce them into the object language [2].

either holds that $C^{\mathcal{I}(w)} = \Delta$ or $C^{\mathcal{I}(w)} = \emptyset$. Technically, such concepts could be used to simulate the context language by associating with every context concept $\boldsymbol{C}$ its global counterpart $C_{\boldsymbol{C}}$, and requiring for every $w \in \mathfrak{W}$ the correspondence: $w \in \boldsymbol{C}^{\mathcal{J}}$ *iff* $C_{\boldsymbol{C}}^{\mathcal{I}(w)} = \Delta$. Given this restriction, concepts $\langle \boldsymbol{r}.\boldsymbol{C}\rangle D$ and $\langle \boldsymbol{C}\rangle D$ of DLCs, could be then roughly translated into $\Diamond_r(C_{\boldsymbol{C}} \sqcap D)$ and $\Diamond(C_{\boldsymbol{C}} \sqcap D)$, of the respective two-dimensional DLs $(\mathbf{K}_n)_{\mathcal{L}}$ and $(\mathbf{S5})_{\mathcal{L}}$. However, even if a complete reduction of this kind was technically possible, this approach would yield a formalism conceptually inadequate to our motivation, as the semantics of global concepts is defined purely in terms of the object domain and not the domain of contexts as we specifically intend here. Moreover, the interaction between the two levels of representation would be highly obscured, making it hard to define fragments of $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ in a modular fashion — simply by selecting DLs of desired expressiveness for $\mathcal{L}_C$ and $\mathcal{L}_O$.

In order to prove decidability of the knowledge base satisfiability problem in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$, we devise a *quasistate elimination algorithm*, similar to the one by Kurucz *et al.* [29, Theorem 6.61], which extends the standard Pratt-style type elimination technique, commonly used in demonstrating upper bounds for modal logics. Essentially, instead of looking directly for a model of a knowledge base, we abstract from the possibly infinite domains $\mathfrak{C}$ and Δ, and consider only a finite number of quasistates which represent possible types of contexts, inhabited by a finite number of possible types of objects. Further, all object types and all quasistates which do not satisfy certain criteria are iteratively eliminated. If at the end of the elimination process there are some non-empty quasistates left, it is guaranteed that a model exists. In the opposite case, the knowledge base is unsatisfiable. As the proof is quite involved we only sketch its key steps below, while full details and missing definitions are presented in the appendix.

Theorem 3 (Upper bound) *Satisfiability of a knowledge base in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$, for $\mathcal{L}_O = \mathcal{L}_C = \mathcal{SHIO}$, any combination of context operators $\mathfrak{F}_1/\mathfrak{F}_2$ and for local interpretation of object roles, is decidable in* 2ExpTime.

Proof sketch. Let $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ be a $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-knowledge base whose satisfiability we want to decide. We assume that several satisfiability preserving transformations are initially applied to $\mathcal{K}$, helping to reduce the number of syntactic cases to be addressed in the proof. We use the following notation to mark the sets of symbols of particular types occurring in $\mathcal{K}$:

- $con_c(\mathcal{K})$: all context concepts, closed under negation,
- $con_o(\mathcal{K})$: all object concepts, closed under negation,
- $sub_o(\mathcal{K})$: all axioms in the set $\{\varphi \mid \boldsymbol{C} : \varphi \in \mathcal{O} \text{ for any } \boldsymbol{C}\}$.

Next, we introduce three central notions: context types, object types and quastistates.

A *context type* for $\mathcal{K}$ is a subset $c \subseteq con_c(\mathcal{K})$, where:

- $\boldsymbol{C} \in c$ iff $\neg\boldsymbol{C} \notin c$, for all $\boldsymbol{C} \in con_c(\mathcal{K})$,

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

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

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

A *quasistate* for $\mathcal{K}$ is a tuple $q = \langle c_q, f_q, O_q \rangle$, where c_q is a context type for $\mathcal{K}$, $f_q \subseteq sub_o(\mathcal{K})$ and O_q is a non-empty set of object types for $\mathcal{K}$.

Intuitively, a context type represents a possible element of the context domain in a $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-model. The precise identity of this element is irrelevant. What matters is only the set of concepts of the context language which could completely describe this element in a model of $\mathcal{K}$, where the context language is restricted only to the concepts (and their negations) occurring in $\mathcal{K}$. Analogically, an object type represents a full description of a possible element of the object domain. Finally, a quasistate captures a "slice" of a model representing one possible context inhabited by a set of possible objects.

Eventually we define the notion of quasimodel, which corresponds to a finitized abstraction of a $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-model. A *quasimodel* for $\mathcal{K}$ is a set $\mathfrak{N}$ of quasistates for $\mathcal{K}$ satisfying a number of specific "integrity" conditions. Most importantly, it has to be guaranteed that all axioms of $\mathcal{K}$ are satisfied by the appropriate types (object and context) in the appropriate quasistates. Also, it has to be ensured that for all types containing concepts based on some forms of existential restrictions ($\exists r.\cdot$, $\exists \boldsymbol{r}.\cdot$, $\langle \boldsymbol{r}.\cdot \rangle$, $\langle \cdot \rangle$) there exist suitable types that could possibly represent their matching successors in a model. Under these constraints we are then able to prove the key quasimodel lemma.

Lemma 1 *There is a quasimodel for* $\mathcal{K}$ iff *there is an* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$*-model of* $\mathcal{K}$.

The basic, brute-force algorithm deciding whether a quasimodel for $\mathcal{K}$ exists starts by enumerating the set $\mathfrak{N}$ of all possible quasistates and then iteratively eliminates all those which violate any of the constraints mentioned above. If the elimination terminates returning a non-empty set of quasistates each containing at least one object type, then this set is guaranteed to be a quasimodel and the search is finished with the answer "$\mathcal{K}$ is satisfiable". Else, no quasimodel exists and the algorithm returns "$\mathcal{K}$ is unsatisfiable".

As the maximum size of a quasimodel (total size of all quasistates) is double exponential in the size of $\mathcal{K}$, and a single run of the elimination procedure cannot take more than a polynomial number of steps in the total size of the initially enumerated quasistates, therefore the algorithm must terminate in at most double exponential time in the size of $\mathcal{K}$. Hence deciding satisfiability of a $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-knowledge base is in 2ExpTime. ❑

Note that this result holds only for local interpretation of object roles. When roles are rigid, the satisfiability problem becomes undecidable, which can be demonstrated by a straightforward reduction of the $\mathbb{N} \times \mathbb{N}$-tiling problem [29].

It turns out that in this setting the 2ExpTime upper bound is optimal, at least whenever context operators $\mathfrak{F}_1$ are involved and $\mathcal{L}_O$ subsumes the DL $\mathcal{ALC}$. The proof relies again on close relationships of CDLs to two-dimensional DLs. Here, we focus on the logic $(\mathbf{DAlt}_n)_\mathcal{L}$ [29] and show that it can be reduced to $(\mathbf{K}_n)_\mathcal{L}$, which in turn, as shown in Theorem 1, can be embedded in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, for $\mathcal{L}_O = \mathcal{L}$. The logic $(\mathbf{DAlt}_n)_\mathcal{L}$ extends a DL $\mathcal{L}$ with a set of functional modalities $\bigcirc_i$, i.e. operators associated with accessibility relations R_i satisfying the properties of seriality ($\mathbf{D}$) and quasi-functionality ($\mathbf{Alt}$):

(seriality) $\forall w \in \mathfrak{W}\ \exists v \in \mathfrak{W}\ (wR_iv)$,

(quasi-functionality) $\forall w, v, u \in \mathfrak{W}\ (wR_iv \wedge wR_iu \to v = u)$.

It is easy to show that there exists a chain of straightforward reductions relating the logics between $(\mathbf{DAlt}_n)_\mathcal{L}$ and $(\mathbf{K}_n)_\mathcal{L}$, including also $(\mathbf{D}_n)_\mathcal{L}$, based on serial frames, and $(\mathbf{Alt}_n)_\mathcal{L}$, based on quasi-functional frames.

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

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

Proof. To see that the reductions indeed hold, it is sufficient to notice that the properties of seriality and quasi-functionality can be axiomatized (or at least emulated) in the languages of the considered logics. Hence, if $(C, \mathcal{T})$ is an instance of the concept satisfiability problem w.r.t. a global TBox in some lefthandside logic, then one can decide it in a righthandside logic by applying simple transformations of C and $\mathcal{T}$ which encode the missing conditions and thus allow for enforcing only models which are bisimilar to those of the original logic:

(seriality) Let $\mathcal{T}' = \mathcal{T} \cup \{\top \sqsubseteq \Diamond_i \top \mid i \in (1, n)\}$, 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.

(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. ❑

Next, we reduce the 2ExpTime-hard *word problem* for exponentially space-bounded *Alternating Turing Machine* [16] to the concept satisfiability problem in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$. For space limitations the full proof is again presented in the appendix.

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

Proof sketch. Let $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$ be an Alternating Turing Machine (ATM), 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 a sequence $\omega q \omega'$, where $\omega\omega'$ is a word based on Σ, q is a state of the machine and the head of the machine is on the leftmost symbol of ω'. A succeeding configuration is defined by transitions δ. An ATM *computation tree* is a finite tree whose nodes are labeled with configurations, where:

- the root contains the *initial configuration* $q_0\omega$, where ω is of length n,
- every configuration $\omega q \omega'$ on the tree, where $\omega\omega'$ is of length at most 2^n, is succeed by:
  - at least one successor configuration, whenever q is an existential state,
  - all successor configurations, whenever q is a universal state,
- 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 ω *iff* there exists an accepting ATM tree with $q_0\omega$ as its initial configuration.

To reduce the word problem, for a word ω over Σ, we formulate a global TBox $\mathcal{T}_\mathcal{M}$ and a concept $C_{\mathcal{M},\omega}$ in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$, such that $\mathcal{M}$ accepts ω *iff* $C_{\mathcal{M},\omega}$ is satisfiable w.r.t. $\mathcal{T}_\mathcal{M}$. The size of the resulting problem instance $(C_{\mathcal{M},\omega}, \mathcal{T}_\mathcal{M})$ is at most polynomial in the size of $\mathcal{M}$ and ω. The reduction is quite involved and essentially relies on an extensive use of **DAlt**-modalities. We define two separate sets of such 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 $q' \in Q\}$,

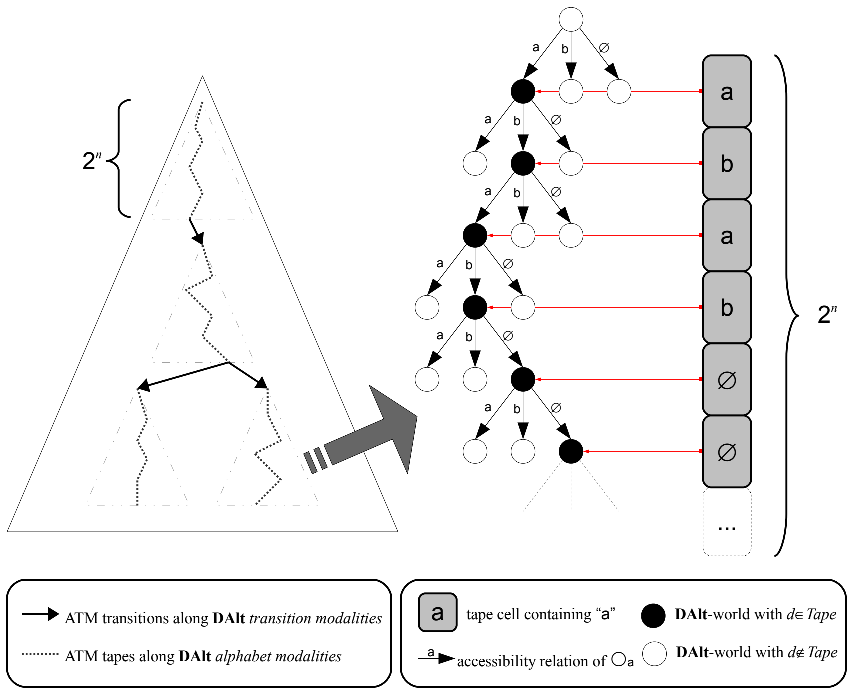

Figure 4: Embedding of ATM computation trees (left) and ATM tapes (right) in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models.

By a suitable use of these operators we are able to encode the complete syntactic structure of an ATM computation tree in the specific fragments of $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models, as illustrated in Figure 4. In particular, a selected object domain individual $d \in \Delta$ is forced to instantiate the designated concept *Tape* exactly in those **DAlt**-worlds which represent the cells of the ATM tape in the subsequent configurations. The accessibility relations connecting those worlds encode the content of the cells and the transitions between the configurations. Further, specific concepts are used to represent the corresponding positions of the head and the states of the machine. Finally, using special counting concepts, which enable traversing the ATM tree structure downwards and upwards, we align the succeeding configurations semantically, ensuring they satisfy the constraints of the respective transitions. ❑

This result grants immediately a lower complexity bound for $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$.

Theorem 5 (Lower bound) *Deciding satisfiability of a knowledge base in* $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$*, for* $\mathcal{L}_O = \mathcal{ALC}$ *and arbitrary* $\mathcal{L}_C$*, with context operators* $\mathfrak{F}_1$ *and for local interpretation of object roles, is* 2ExpTime*-hard.*

Proof. Immediate by Proposition 1 and Theorems 1 and 4. ❑

As an interesting corollary, we also obtain a lower bound for the problem of concept satisfiability w.r.t. global TBoxes in several two-dimensional DLs considered above, most prominently in $(\mathbf{K}_n)_{\mathcal{ALC}}$.

Corollary 1 *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*-hard.*

Proof. Immediate by Proposition 1 and Theorem 4. ❑

This increase in the complexity by one exponential, as compared to $\mathcal{ALC}$ alone (for which the problem is ExpTime-complete [5]), is notable and quite surprising. It could be expected that without rigid roles the satisfiability problem in two-dimensional DLs can be reduced to satisfiability in fusion models of conventional DLs [29]. This in turn should yield ExpTime upper bound by means of the standard techniques. However, as the following counterexample for $(\mathbf{K}_n)_{\mathcal{ALC}}$ shows, 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. This effect, amplified by presence of multiple modalities and global TBoxes (which can enforce infinite **K**-trees), makes the reasoning harder. The result 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.

Both complexity bounds established above warrant the following final conclusions, where by $\preceq$ we denote the relation of being less or equally expressive.

Theorem 6 *Deciding satisfiability of a knowledge base in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, for $\mathcal{L}_C \preceq \mathcal{SHIO}$, $\mathcal{ALC} \preceq \mathcal{L}_O \preceq \mathcal{SHIO}$, with at least context operators $\mathfrak{F}_1$ (and possibly also $\mathfrak{F}_2$) and for local interpretation of object roles, is* 2ExpTime*-complete.*

Proof. Immediate by Theorem 3 and Theorem 5. ❑

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

Proof. Immediate by Proposition 1, Corollary 1 and Theorems 3 and 1. ❑

7 Related work

The problem of formalizing contexts in AI has been commonly studied in the literature (see [1] for an overview). McCarthy's theory, exploited in this paper, has been translated into a number of logic systems [15, 13, 36], typically of a strongly modal flavor. For instance, in the propositional logic of context [15] an assertion $ist(\boldsymbol{c}, \varphi)$ can be restated as a modal formula $\Box_{\boldsymbol{c}}\varphi$, where the behavior of $\Box_{\boldsymbol{c}}$ is suitably axiomatized in order to capture possibly many context-based operations, e.g., entering and exiting contexts, lifting knowledge from one context to another, etc. As opposed to those traditional approaches, we do not go deep into axiomatizing specific properties and mechanisms of our context logic, but rather advocate straightforward use of the standard two-dimensional semantics and two-sorted languages in order to offer a minimal, yet highly flexible framework for declarative modeling of contextualized knowledge.

Another dominant tradition in the field originates from the paradigm of *multi-context logics* (MCLs), introduced by Giunchiglia *et al.* [18, 19, 17]. While most formalisms based on McCarthy's theory support mainly the divide-and-conquer scenarios, the MCLs are naturally tailored to compose-and-conquer applications [10], focusing on the mechanisms of bridging multiple local representations. Again, we do not enforce any such particular mechanism, compromising application-driven functionalities of DLCs for their generality.

These two perspectives on operationalizing contexts in knowledge systems can be further linked to two areas of research within the field of DLs: two-dimensional DLs and logic-based ontology integration. Two-dimensional DLs [43], addressed at length in this paper, are quintessential for this work as they constitute a well-grounded paradigm of constructing logics for modeling knowledge relative to implicit semantic states, regardless of the particular philosophical nature of those states. On the syntactic level, DLCs differ from the standard two-dimensional DLs in a number of specific aspects, predominantly, in the involvement of a second sort of language for speaking about the second semantic dimension. To our knowledge such an extension has not yet been studied in the literature. The area of logic-based ontology integration focuses on the problem of integrating knowledge contained in multiple, independent sources (DL-based ontologies). Among many existing solutions there are Package-based DLs [8], Distributed DLs [9], $\mathcal{E}$-Connections [30], semantic imports [37], and others. Each offers a formal mechanism of relating the vocabularies belonging to different sources, while to a large extent preserving the semantic independence of those sources. To this end the formalisms employ certain semantic relations for linking models of the respective ontologies. As shown in Section 5, DLCs can also support such scenarios by grounding the integration mechanism in the possible world semantics. Naturally, all those approaches should likely exhibit different formal properties and practical behavior, whose precise characterization would require a separate study.

The need for explicit treatment of context has been also broadly acknowledged by the Semantic Web community and introduced in diverse aspects over different Semantic Web knowledge representation frameworks [11, 23, 12, 7, 39, 41, 45]. Particularly interesting here are the efforts on representing and reasoning with meta-level descriptions of knowledge, such as involved in our axioms of type $\boldsymbol{C}{:}\varphi$, where $\boldsymbol{C}$ can be seen as a meta-level annotation of the assertion φ. The framework presented in [41] supports simple annotations over OWL axioms and further allows for selection of those axioms based on annotation queries. A similar approach, proposed in [45], considers arbitrary Semantic Web data described with annotations belonging to certain well-behaved annotation languages, e.g., temporal or fuzzy, and supports some basic forms of annotation-driven inference over the data. The main shortcoming of those and similar contributions is, in our view, a quite limited treatment of the meta-level representation, which is often expressed in restricted, non-logical languages, impeding the semantic transparency and reasoning capabilities of the systems.

Finally, we acknowledge the substantial work by Serafini and Homola [39], in which the framework of Contextualized Knowledge Repositories (CKRs) is defined. Notably, this proposal incorporates the key components integral to our DLCs: DL-based representation of object knowledge, contexts as formal objects, a two-dimensional semantics, a mechanism of relating knowledge from different contexts, meta-level descriptions of contexts. However, apart from the full use of DL languages over the object dimension, the remaining features of CKRs are somewhat restricted compared to DLCs. The number of contexts in CKRs is always finite, the mechanism for relating knowledge from different contexts operates on one-to-one basis, and finally, the context language does not embrace the full expressiveness of standard DL languages, but instead supports descriptions of contexts in terms of property-value pairs. Consequently, DLCs, although closely aligned with CKRs in terms of the motivation, conceptual foundations, and basic formal insights, can be seen as a formal generalization of this framework, characterized by a broader and more liberal view on the context-level components of the architecture. Notably, this generality of DLCs is penalized with the high computational complexity of the most expressive fragment. However, it is a trade-off that can be in principle controlled by additional syntactic restrictions, suitably taming the interaction between the two dimensions. On the contrary, CKRs do not tend to increase the complexity of reasoning compared to that of the underlying DL object language, but in turn, leave less space for adjusting the contextualization mechanism.

8 Conclusions

Representing inherently contextualized knowledge, as well as reasoning with multiple, heterogenous, but semantically interoperating knowledge sources, are both interesting and practically vital problems within the area of the DL-based knowledge representation. It is our strong belief that these two challenges are in fact two sides of the same coin and, consequently, they should be approached within the same, unifying formal framework. In this paper, we have proposed such a framework, founded on a novel family of two-dimensional, two-sorted *Description Logics of Context*, which arguably supports both functionalities, seamlessly integrated on the grounds of one formal theory. The pivotal premise of this theory is that contexts should be interpreted as possible worlds in the second modal dimension, added to the standard semantics of DLs. In this way the instrumental, application-agnostic spirit of McCarthy's theory of contexts can be successfully combined with the formal machinery of modal logics.

The work presented in this paper establishes the generic foundations for the DLC framework and opens up a number of theoretical and practical problems which should be addressed in the future research. One important direction is to investigate how different notions common to traditional context-based systems (e.g., managing local inconsistencies, representing the generality hierarchy of contexts, etc.) can be effectively restated within DLCs. Another course of research should be dedicated to identification and formal analysis of specific fragments of the framework that could be especially useful in practice, particularly considering Semantic Web applications. For instance, a scenario of integrating a finite number of ontologies does not in principle require the full expressiveness of DLCs. Similarly, an efficient support for reasoning with contextually annotated Semantic Web data could be likely provided via a more lightweight fragment. Finally, on a more abstract level, it could be interesting to investigate whether a similar methodology of constructing two-dimensional, two-sorted formalisms could be applicable to combinations of DLs with other modal logics, e.g. spatial or temporal, in order to support fine-grained descriptions of the second semantic dimension by means of a dedicated vocabulary.

Acknowledgements

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

References

[1] Varol Akman and Mehmet Surav. Steps toward formalizing context. *AI Magazine*, 17:55–72, 1996.

[2] Carlos Areces and Maarten de Rijke. From description to hybrid logics, and back. In *Proc. of Advances in Modal Logic (AML-01)*. 2001.

[3] Alessandro Artale, Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporalising tractable description logics. In *Proc. of the International Symposium on Temporal Representation and Reasoning (TIME-07)*, 2007.

[4] Alessandro Artale, Carsten Lutz, and David Toman. A description logic of change. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI-07)*, 2007.

[5] Franz Baader, Diego Calvanese, Deborah L. Mcguinness, Daniele Nardi, and Peter F. Patel-Schneider. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, 2003.

[6] Franz Baader, Ian Horrocks, and Ulrike Sattler. Description logics as ontology languages for the semantic web. In Dieter Hutter and Werner Stephan, editors, *Mechanizing Mathematical Reasoning: Essays in Honor of Jörg Siekmann on the Occasion of His 60th Birthday*, number 2605 in Lecture Notes in Artificial Intelligence, pages 228–248. Springer, 2005.

[7] Jie Bao, Jiao Tao, Deborah L McGuinness, and Paul Smart. Context representation for the semantic web. *Proc. of Web Science Conference*, 2010.

[8] Jie Bao, George Voutsadakis, Giora Slutzki, and Vasant Honavar. Package-based description logics. In Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors, *Modular Ontologies*, pages 349–371. Springer, 2009.

[9] Alex Borgida and Luciano Serafini. Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantics*, 1:153–184, 2003.

[10] Paolo Bouquet, Chiara Ghidini, Fausto Giunchiglia, and Enrico Blanzieri. Theories and uses of context in knowledge representation and reasoning. *Journal of Pragmatics*, 35(3):455 – 484, 2003.

[11] Paolo Bouquet, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. C-OWL: Contextualizing ontologies. In Dieter Fensel, Katia P. Sycara, and John Mylopoulos, editors, *International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pages 164–179. Springer, 2003.

[12] Paolo Bouquet, Luciano Serafini, and Heiko Stoermer. Introducing context into rdf knowledge bases. In *Proc. of the Italian Semantic Web Workshop (SWAP-05)*, 2005.

[13] Saša Buvač. Quantificational logic of context. In *Proc. of the Conference on Artificial Intelligence (AAAI-96)*, 1996.

[14] Saša Buvač, Vanja Buvac, and Ian A. Mason. Metamathematics of contexts. *Fundamenta Informaticae*, 23:412–419, 1995.

[15] Saša Buvač and Ian A. Mason. Propositional logic of context. In *Proc. of the Conference on Artificial Intelligence (AAAI-93)*, 1993.

[16] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, 1981.

[17] Chiara Ghidini and Fausto Giunchiglia. Local models semantics, or contextual reasoning=locality+compatibility. *Artificial Intelligence*, 127(2):221 – 259, 2001.

[18] Fausto Giunchiglia. Contextual reasoning. *Epistemologia*, XVI:345–364, 1993.

[19] Fausto Giunchiglia and Luciano Serafini. Multilanguage hierarchical logics, or: How we can do without modal logics. *Artificial Intelligence*, 65(1):29 – 70, 1994.

[20] Krzysztof Goczyla, Wojciech Waloszek, and Aleksander Waloszek. Contextualization of a DL knowledge base. In *Proc. of the International Workshop on Description Logics (DL-07)*, 2007.

[21] Davide Grossi. *Desigining Invisible Handcuffs. Formal Investigations in Institutions and Organizations for Multi-Agent Systems.* PhD thesis, Utrecht University, 2007.

[22] Ramanathan V. Guha. *Contexts: a formalization and some applications.* PhD thesis, Stanford University, 1991.

[23] Ramanathan V. Guha, Rob McCool, and Richard Fikes. Contexts for the semantic web. In *Proc. of the International Semantic Web Conference (ISWC-04)*, 2004.

[24] Jan Hladik and Jörg Model. Tableau systems for SHIO and SHIQ. In *Proc. of the International Workshop on Description Logics (DL-04)*, 2004.

[25] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The Even More Irresistible SROIQ. In *Proc. of the International COnference on Principles of Knowledge Representation and Reasoning (KR-06).* AAAI Press, 2006.

[26] Zhisheng Huang and Heiner Stuckenschmidt. Reasoning with multi-version ontologies: A temporal logic approach. In *Proc. of the Intenational Semantic Web Conference (ISWC-05).* 2005.

[27] Szymon Klarman and Víctor Gutiérrez-Basulto. $\mathcal{ALC}_{\mathcal{ALC}}$: A context description logic. In *Proc. of the European Conference on Logics in Artificial Intelligence (JELIA-10).* 2010.

[28] Szymon Klarman and Víctor Gutiérrez-Basulto. Two-dimensional description logics for context-based semantic interoperability. In *Proc. of the Conference on Artificial Intelligence (AAAI-11).* 2011.

[29] Agi Kurucz, Frank Wolter, Michael Zakharyaschev, and Dov M. Gabbay. *Many-Dimensional Modal Logics: Theory and Applications.* Elsevier, 2003.

[30] Oliver Kutz, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. $\mathcal{E}$-connections of description logics. In *Proc. of the International Workshop on Description Logics (DL-03)*, 2003.

[31] Doug Lenat. The dimensions of context space. Technical report, CYCORP, 1998.

[32] Carsten Lutz and Lutz Schröder. Probabilistic description logics for subjective uncertainty. In *Proc. of the International Conference on Principles of Knowledge Representation and Reasoning (KR-10)*, 2010.

[33] Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporal description logics: A survey. In *Proc. of the International Symposium on Temporal Representation and Reasoning (TIME-08)*, 2008.

[34] John McCarthy. Generality in artificial intelligence. *Communications of the ACM*, 30:1030–1035, 1987.

[35] John McCarthy. Notes on formalizing context. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI-93)*, 1993.

[36] Rolf Nossum. A decidable multi-modal logic of context. *Journal of Applied Logic*, 1(1-2):119 – 133, 2003.

[37] Jeff Z. Pan, Luciano Serafini, and Yuting Zhao. Semantic import: an approach for partial ontology reuse. In *Proc. of the Workshop on Modular Ontologies (WoMO-06)*, 2006.

[38] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[39] Luciano Serafini and Martin Homola. Contextualized knowledge repositories for the semantic web. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 12:64–87, 2012.

[40] Kent Spackman. SNOMED CT style guide: Situations with explicit context. Technical report, SNOMED CT, 2008.

[41] Thanh Tran, Peter Haase, Boris Motik, Bernardo Cuenca Grau, and Ian Horrocks. Metalevel information in ontology-based applications. In *Proc. of the Conference on Artificial Intelligence (AAAI-08)*, 2008.

[42] Frank Wolter and Michael Zakharyaschev. Satisfiability problem in description logics with modal operators. In *Proc. of the International Conference on Principles of Knowledge Representation and Reasoning (KR-98)*, 1998.

[43] Frank Wolter and Michael Zakharyaschev. Multi-dimensional description logics. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1999.

[44] Frank Wolter and Michael Zakharyaschev. Dynamic description logics. In *Proc. of Advances in Modal Logic (AiML-98)*, 2000.

[45] Antoine Zimmermann, Nuno Lopes, Axel Polleres, and Umberto Straccia. A general framework for representing, reasoning and querying with annotated Semantic Web data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 11:72–95, 2012.

Appendix

Below we present full proofs of two key results sketched in Section 6.

Upper complexity bound

First we demonstrate decidability and the implied 2ExpTime upper complexity bound for the knowledge base satisfiability problem in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$. For the proof we devise a *quasistate elimination algorithm*, similar to the one by Kurucz *et al.* [29, Theorem 6.61], which extends the standard Pratt-style type elimination technique, commonly used in demonstrating upper bounds for modal logics. Essentially, instead of looking directly for a model of a knowledge base, we abstract from the possibly infinite domains $\mathfrak{C}$ and Δ, and consider only a finite number of quasistates which represent possible types of contexts, inhabited by a finite number of possible types of objects. Further, all object types and all quasistates which do not satisfy certain criteria are iteratively eliminated. If at the end of the elimination process there are some non-empty quasistates left, it is guaranteed that a model exists. Otherwise, the knowledge base is unsatisfiable.

Theorem 3 (Upper bound) *Satisfiability of a knowledge base in* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$*, for* $\mathcal{L}_O = \mathcal{L}_C = \mathcal{SHIO}$*, any combination of context operators* $\mathfrak{F}_1/\mathfrak{F}_2$ *and for local interpretation of object roles, is decidable in* 2ExpTime.

Proof. Let $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ be a knowledge base in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$, for $\mathcal{L}_O = \mathcal{L}_C = \mathcal{SHIO}$ with context operators $\mathfrak{F}_1$ and $\mathfrak{F}_2$.

By $\cdot^-$ we denote the inverse constructor for roles and assume that $(r^-)^- = r$ (resp. $(\boldsymbol{r}^-)^- = \boldsymbol{r}$). Let f be a set of $\mathcal{SHIO}$ axioms. Then by $\sqsubseteq_f^*$ we denote the reflexive-transitive closure of $\sqsubseteq$ on $\{r \sqsubseteq s, s^- \sqsubseteq r^- \mid r \sqsubseteq s \in f, \text{ for } r, s \in N_R\}$ (resp. $\{\boldsymbol{r} \sqsubseteq \boldsymbol{s}, \boldsymbol{s}^- \sqsubseteq \boldsymbol{r}^- \mid \boldsymbol{r} \sqsubseteq \boldsymbol{s} \in f, \text{ for } \boldsymbol{r}, \boldsymbol{s} \in M_R\}$). W.l.o.g. we assume that none of the constructors $[\boldsymbol{r}.\cdot]$, $[\cdot]$, $\forall r.\cdot$, $\forall \boldsymbol{r}.\cdot$, $\sqcup$ occur in $\mathcal{K}$. Further, all axioms $\boldsymbol{a} : \varphi \in \mathcal{O}$ are restated in the equivalent form $\{a\} : \varphi$ and all the following types of (sub)formulas are replaced with their equivalents:

$$
\begin{array}{lcl}
C(a) & \Rightarrow & \{a\} \sqsubseteq C, \\
r(a,b) & \Rightarrow & \{a\} \sqsubseteq \exists r.\{b\}, \\
\boldsymbol{C}(\boldsymbol{a}) & \Rightarrow & \{\boldsymbol{a}\} \sqsubseteq \boldsymbol{C}, \\
\boldsymbol{r}(\boldsymbol{a},\boldsymbol{b}) & \Rightarrow & \{\boldsymbol{a}\} \sqsubseteq \exists \boldsymbol{r}.\{\boldsymbol{b}\}.
\end{array}
$$

The following notation is used to mark the sets of symbols of particular type occurring in $\mathcal{K}$:

$con_c(\mathcal{K})$:	all context concepts, closed under negation,
$con_o(\mathcal{K})$:	all object concepts, closed under negation,
$rol_c(\mathcal{K})$:	all context roles,
$rol_c^+(\mathcal{K}) \subseteq rol_c(\mathcal{K})$:	all transitive context roles,
$rol_o(\mathcal{K})$:	all object roles,
$rol_o^+(\mathcal{K}) \subseteq rol_o(\mathcal{K})$:	all transitive object roles,

$ind_c(\mathcal{K})$:	all context individual names,
$ind_o(\mathcal{K})$:	all object individual names,
$sub_o(\mathcal{K})$:	all axioms from $\{\varphi \mid \boldsymbol{C} : \varphi \in \mathcal{O} \text{ for any } \boldsymbol{C}\}$.

First, we define three types of entities: context types, object types and quasistates, which are the basic building blocks for the finite representations of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-models, called quasimodels. Intuitively, a context type represents a possible element of the context domain in a $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model. The precise identity of this element is irrelevant. What matters is only the set of concepts of the context language which could completely describe this element in a model of $\mathcal{K}$, where the context language is restricted only to the concepts (and their negations) occurring in $\mathcal{K}$. Analogically, an object type represents a full description of a possible element of the object domain. Finally, a quasistate captures a "slice" of a model representing one possible context inhabited by a set of possible objects.

A *context type* for $\mathcal{K}$ is a subset $c \subseteq con_c(\mathcal{K})$, where:

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

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

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

Definition 7 (matching object role-successor) *Let t, t' be two object types for $\mathcal{K}$ and $f \subseteq \mathrm{sub}_o(\mathcal{K})$. For any $r \in rol_o(\mathcal{K})$, t' is a* matching r-successor *for t under f* iff *the following conditions are satisfied:*

- $\{\neg C \mid \neg\exists r.C \in t\} \subseteq t'$ *and* $\{\neg C \mid \neg\exists r^-.C \in t'\} \subseteq t$,
- *if* $r \in rol_o^+(\mathcal{K})$ *then* $\{\neg\exists r.C \in t\} \subseteq t'$ *and* $\{\neg\exists r^-.C \in t'\} \subseteq t$,
- *t' is a matching s-successor for t under f, for every $s \in rol_o(\mathcal{K})$ such that $r \sqsubseteq_f^* s$,*
- *t is a matching s-successor for t' under f, for every $s \in rol_o(\mathcal{K})$ such that $r^- \sqsubseteq_f^* s$.*

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

(qS) if $\exists r.D \in t$ then t has a matching r-successor $t' \in O_q$ under f_q.

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

(qC1) for every $a \in ind_o(\mathcal{K})$ there exists a unique $t \in O_q$ such that $\{a\} \in t$,

(qC2) for every $\boldsymbol{C} : \varphi \in \mathcal{O}$, if $\boldsymbol{C} \in c_q$ then $\varphi \in f_q$,

(qC3) for every $C \sqsubseteq D \in f_q$ and $t \in O_q$, if $C \in t$ then $D \in t$,

(qC4) for every $t \in O_q$ and $\neg\langle \boldsymbol{C} \rangle D \in t$, if $\boldsymbol{C} \in c_q$ then $\neg D \in t$.

A *linkage* between two quasistates $q = \langle c_q, f_q, O_q \rangle$ and $q' = \langle c'_q, f'_q, O'_q \rangle$ for $\mathcal{K}$ is a mapping $\lambda = g \cup h$, where $g : O_q \mapsto O'_q$ and $h : O'_q \mapsto O_q$, such that for every $a \in ind_o(\mathcal{K})$ and $t \in O_q \cup O'_q$, $\{a\} \in t$ *iff* $\{a\} \in \lambda(t)$.

Definition 8 (matching $\mathfrak{F}_2$-successor) *Let $q = \langle c_q, f_q, O_q \rangle$ and $q' = \langle c'_q, f'_q, O'_q \rangle$ be two quasistates for $\mathcal{K}$. Then q' is a* matching $\mathfrak{F}_2$-successor *for q via a linkage λ* iff *for every $t \in O_q \cup O'_q$, $\{\langle \boldsymbol{C} \rangle D \in t\} \cup \{\neg\langle \boldsymbol{C} \rangle D \in t\} = \{\langle \boldsymbol{C} \rangle D \in \lambda(t)\} \cup \{\neg\langle \boldsymbol{C} \rangle D \in \lambda(t)\}$.*

Definition 9 (matching $\mathfrak{F}_1$-successor) *Let $q = \langle c_q, f_q, O_q \rangle$ and $q' = \langle c'_q, f'_q, O'_q \rangle$ be two quasistates for $\mathcal{K}$. For any $\boldsymbol{r} \in rol_c(\mathcal{K})$, q' is a* matching $\boldsymbol{r}$-successor *for q via a linkage λ* iff *q' is a matching $\mathfrak{F}_2$-successor for q via λ and the following conditions are satisfied:*

- $\{\neg \boldsymbol{C} \mid \neg\exists \boldsymbol{r}.\boldsymbol{C} \in c_q\} \subseteq c'_q$ *and* $\{\neg \boldsymbol{C} \mid \neg\exists \boldsymbol{r}^-.\boldsymbol{C} \in c'_q\} \subseteq c_q$,
- *if* $\boldsymbol{r} \in rol_c^+(\mathcal{K})$ *then* $\{\neg\exists \boldsymbol{r}.\boldsymbol{C} \in c_q\} \subseteq c'_q$ *and* $\{\neg\exists \boldsymbol{r}^-.\boldsymbol{C} \in c'_q\} \subseteq c_q$,
- *for every* $t \in O_q$ *and* $t' \in O'_q$, $\{\neg D \mid \neg\langle \boldsymbol{r}.\boldsymbol{C} \rangle D \in t, \boldsymbol{C} \in c'_q\} \subseteq \lambda(t)$, $\{\neg D \mid \neg\langle \boldsymbol{r}.\boldsymbol{C} \rangle D \in \lambda(t'), \boldsymbol{C} \in c'_q\} \subseteq t'$, $\{\neg D \mid \neg\langle \boldsymbol{r}^-.\boldsymbol{C} \rangle D \in \lambda(t), \boldsymbol{C} \in c_q\} \subseteq t$ *and* $\{\neg D \mid \neg\langle \boldsymbol{r}^-.\boldsymbol{C} \rangle D \in t', \boldsymbol{C} \in c_q\} \subseteq \lambda(t')$,
- *for every* $t \in O_q$ *and* $t' \in O'_q$, *if* $\boldsymbol{r} \in rol_c^+(\mathcal{K})$ *then* $\{\neg\langle \boldsymbol{r}.\boldsymbol{C} \rangle D \in t\} \subseteq \lambda(t)$, $\{\neg\langle \boldsymbol{r}.\boldsymbol{C} \rangle D \in \lambda(t')\} \subseteq t'$, $\{\neg\langle \boldsymbol{r}^-.\boldsymbol{C} \rangle D \in \lambda(t)\} \subseteq t$ *and* $\{\neg\langle \boldsymbol{r}^-.\boldsymbol{C} \rangle D \in t'\} \subseteq \lambda(t)$,
- *q' is a matching $\boldsymbol{s}$-successor for q via λ for every $\boldsymbol{s} \in rol_c(\mathcal{K})$ such that $\boldsymbol{r} \sqsubseteq^*_{\mathcal{C}} \boldsymbol{s}$,*
- *q is a matching $\boldsymbol{s}$-successor for q' via λ for every $\boldsymbol{s} \in rol_c(\mathcal{K})$ such that $\boldsymbol{r}^- \sqsubseteq^*_{\mathcal{C}} \boldsymbol{s}$.*

A set of quasistates Q is *saturated iff* for every quasistate $q \in Q$, with $q = \langle c_q, f_q, O_q \rangle$:

(QS1) for every $\exists \boldsymbol{r}.\boldsymbol{C} \in c_q$ there is a matching $\boldsymbol{r}$-successor for q in Q via some linkage λ,

(QS2) for every $t \in O_q$ and $\langle \boldsymbol{C} \rangle D \in t$ there is a matching $\mathfrak{F}_1$-successor $q' = \langle c'_q, f'_q, O'_q \rangle$ for q in Q via some linkage λ, such that $\boldsymbol{C} \in c'_q$ and $D \in \lambda(t)$,

(QS3) for every $t \in O_q$ and $\langle \boldsymbol{r}.\boldsymbol{C} \rangle D \in t$ there is a matching $\boldsymbol{r}$-successor $q' = \langle c'_q, f'_q, O'_q \rangle$ for q in Q via some linkage λ, such that $\boldsymbol{C} \in c'_q$ and $D \in \lambda(t)$.

A *quasimodel* $\mathfrak{N}$ for $\mathcal{K}$ is a non-empty, saturated set of saturated and coherent quasistates for $\mathcal{K}$ satisfying the following conditions:

(M1) for every $\boldsymbol{c} \in ind_c(\mathcal{K})$ there is a unique $q \in \mathfrak{N}$, with $q = \langle c_q, f_q, O_q \rangle$, such that $\{\boldsymbol{c}\} \in c_q$,

(M2) for every $\boldsymbol{C} \sqsubseteq \boldsymbol{D} \in \mathcal{C}$ and $q \in \mathfrak{N}$, with $q = \langle c_q, f_q, O_q \rangle$, if $\boldsymbol{C} \in c_q$ then $\boldsymbol{D} \in c_q$.

We can now prove the quasimodel lemma.

Lemma 1 *There is a quasimodel for $\mathcal{K}$* iff *there is an $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model of $\mathcal{K}$.*

Proof. The key observation which we exploit in this proof is that the constraints (QS1)-(QS3) imposed on quasimodels ensure existence of certain specific quasistates, which represent successors in the context dimension, and existence of special linkage relations allowing for a proper choice of types for the same object in different contexts. To ease reference to these elements we amend the corresponding conditions with the following naming conventions:

(QS1*) in such case call q' a *witness* for $(\exists \boldsymbol{r}.\boldsymbol{C}, q)$ and a linkage λ, enforced by the condition, a *witnessing linkage*,

(QS2*) in such case call q' a *witness* for $(\langle \boldsymbol{C} \rangle D, t, q)$ and a linkage λ, enforced by the condition, a *witnessing linkage*,

(QS3*) in such case call q' a *witness* for $(\langle \boldsymbol{r}.\boldsymbol{C} \rangle D, t, q)$ and a linkage λ, enforced by the condition, a *witnessing linkage*.

($\Rightarrow$) Suppose $\mathfrak{N}$ is a quasimodel for $\mathcal{K} = (\mathcal{C}, \mathcal{O})$. We construct an $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model $\mathfrak{M} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i \in \mathfrak{C}})$ of $\mathcal{K}$ based on $\mathfrak{N}$. We start by defining an interpretation of the context dimension $(\mathfrak{C}, \cdot^{\mathcal{J}})$. First, for every $\boldsymbol{c} \in ind_c(\mathcal{K})$ and $q \in \mathfrak{N}$ such that $\{\boldsymbol{c}\} \in q$, add q to $\mathfrak{C}$ and set $\boldsymbol{c}^{\mathcal{J}} = q$. In case $ind_c(\mathcal{K}) = \emptyset$ set $\mathfrak{C} = \{q\}$ for any $q \in \mathfrak{N}$. Then iteratively extend $(\mathfrak{C}, \cdot^{\mathcal{J}})$ as follows. For every $q \in \mathfrak{C}$, with $q = \langle c_q, f_q, O_q \rangle$:

- for every $\exists \boldsymbol{r}.\boldsymbol{C} \in c_q$ pick a witness q' for $(\exists \boldsymbol{r}.\boldsymbol{C}, q)$ from $\mathfrak{N}$, add it to $\mathfrak{C}$ and set $(q, q') \in \boldsymbol{r}^{\mathcal{J}}$,
- for every $t \in O_q$ and $\langle \boldsymbol{C} \rangle D \in t$ pick a witness q' for $(\langle \boldsymbol{C} \rangle D, t, q)$ from $\mathfrak{N}$ and add it to $\mathfrak{C}$,
- for every $t \in O_q$ and $\langle \boldsymbol{r}.\boldsymbol{C} \rangle D \in t$ pick a witness q' for $(\langle \boldsymbol{r}.\boldsymbol{C} \rangle D, t, q)$ from $\mathfrak{N}$, add it to $\mathfrak{C}$ and set $(q, q') \in \boldsymbol{r}^{\mathcal{J}}$.

Further, we extend the interpretation of roles by iteratively saturating the following steps. For every $q, q', q'' \in \mathfrak{C}$ and $\boldsymbol{r}, \boldsymbol{s} \in rol_c(\mathcal{K})$:

- if $(q, q') \in \boldsymbol{r}^{\mathcal{J}}$ then set $(q', q) \in (\boldsymbol{r}^-)^{\mathcal{J}}$,
- if $(q, q') \in \boldsymbol{r}^{\mathcal{J}}$ and $\boldsymbol{r} \sqsubseteq^*_{\mathcal{C}} \boldsymbol{s}$ then set $(q, q') \in \boldsymbol{s}^{\mathcal{J}}$,
- if $\boldsymbol{r} \in rol^+_c(\mathcal{K})$ and $(q, q'), (q', q'') \in \boldsymbol{r}^{\mathcal{J}}$ then set $(q, q'') \in \boldsymbol{r}^{\mathcal{J}}$.

Finally, for every $\boldsymbol{A} \in con_c(\mathcal{K})$ set $\boldsymbol{A}^{\mathcal{J}} = \{q \in \mathfrak{C} \mid \boldsymbol{A} \in c_q\}$.

By structural induction it follows that all complex context concepts are satisfied by $\mathfrak{M}$ in the expected contexts. In particular, all role restrictions must be satisfied due to an adequate interpretation of context roles, ensuring that:

- role names and their inverses are interpreted as relations which are inverses of each other,
- transitive roles are interpreted as transitive relations,
- the role hierarchies entailed by $\mathcal{C}$ are respected.

Above properties are guaranteed by Definition 9 and the construction of the model. Consequently, since $\mathfrak{N}$ satisfies conditions (M1), (M2), all axioms from the context knowledge base $\mathcal{C}$ must be satisfied. Next we 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 in conditions (QS1)-(QS3). Note that the way the interpretation $(\mathfrak{C}, \cdot^{\mathcal{J}})$ is constructed ensures that for every two contexts 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 $q, q' \in \mathfrak{C}$, with $q = \langle c_q, f_q, O_q\rangle$ and $q' = \langle c'_q, f'_q, O'_q\rangle$ and λ being the witnessing linkage between q and q':

- for every $a \in ind_o(\mathcal{K})$, there is exactly one run $\rho_{a,q} \in \mathfrak{R}$ such that $\{a\} \in \rho_{a,q}(q)$,
- for every $\rho \in \mathfrak{R}$, $\lambda(\rho(q)) = \rho(q')$,
- for every $t \in O_q$ and $t' \in O'_q$, if $\lambda(t) = t'$ then there exists $\rho \in \mathfrak{R}$, such that $\rho(q) = t$ and $\rho(q') = t'$.

We let $\Delta = \mathfrak{R}$, for a coherent set of runs $\mathfrak{R}$ through $\mathfrak{C}$, and for every $q \in \mathfrak{C}$, with $q = \langle c_q, f_q, O_q\rangle$, we fix the corresponding interpretation function $\cdot^{\mathcal{I}(q)}$ as follows:

- for every individual name $a \in ind_o(\mathcal{K})$ set $a^{\mathcal{I}(q)} = \rho_{a,q}(q)$,
- for every concept name $A \in con_o(\mathcal{K})$ set $A^{\mathcal{I}(q)} = \{\rho \in \mathfrak{R} \mid A \in \rho(q)\}$,
- for every role $r \in rol_o(\mathcal{K})$, $\rho \in \mathfrak{R}$ and $\exists r.D \in \rho(q)$ pick $\rho' \in \mathfrak{R}$ such that $\rho'(q)$ is a matching r-successor for $\rho(q)$ under f_q and set $(\rho, \rho') \in r^{\mathcal{I}(q)}$.

Note that by aligning runs with the witnessing linkages we automatically ensure that each object obtains compatible interpretations in every two related contexts. In particular, whenever $d \in (\langle \boldsymbol{r}.\boldsymbol{C}\rangle D)^{\mathcal{I}(q)}$ for some $d \in \Delta$ and $q \in \mathfrak{C}$, there has to exist a context $q' \in \boldsymbol{C}^{\mathcal{J}}$ accessible from q through $\boldsymbol{r}$ in which

$d \in D^{\mathcal{I}(q')}$. By the same token, whenever $d \in (\langle \boldsymbol{C} \rangle D)^{\mathcal{I}(q)}$, there must be a context $q' \in \boldsymbol{C}^{\mathcal{J}}$ such that $d \in D^{\mathcal{I}(q')}$.

Further, as before, we extend the interpretation of roles by iteratively saturating the following steps. For every $q \in \mathfrak{C}$, with $q = \langle c_q, f_q, O_q \rangle$, every $\rho, \rho', \rho'' \in \mathfrak{R}$ and $r, s \in rol_o(\mathcal{K})$:

- if $(\rho, \rho') \in r^{\mathcal{I}(q)}$ then set $(\rho', \rho) \in (r^-)^{\mathcal{I}(q)}$,
- if $(\rho, \rho') \in r^{\mathcal{I}(q)}$ and $r \sqsubseteq^*_{f_q} s$ then set $(\rho, \rho') \in s^{\mathcal{I}(q)}$,
- if $r \in rol_o^+(\mathcal{K})$ and $(\rho, \rho'), (\rho, \rho'') \in r^{\mathcal{I}(q)}$ then set $(\rho, \rho'') \in r^{\mathcal{I}(q)}$.

Similarly as in the context dimension, Definition 7 along with way the model is constructed ensure an adequate interpretation of all roles. Consequently, by structural induction it is not difficult to see that all object concepts are satisfied by $\mathfrak{M}$ as expected and thus, since $\mathfrak{N}$ satisfies conditions (qC1)-(qC4), all axioms from the object knowledge base $\mathcal{O}$ must be also satisfied.

($\Leftarrow$) This direction is straightforward. Let $\mathfrak{M} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i \in \mathfrak{C}})$ be a $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model of $\mathcal{K}$. We construct a quasimodel $\mathfrak{N}$ 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. We can then define a quasistate for every $c \in \mathfrak{C}$ as $q_c = \langle t_c, f_c, O_c \rangle$, where $\mathbf{t}(c) = \langle t_c, f_c \rangle$. Finally, let $\mathfrak{N} = \{q_c \mid c \in \mathfrak{C}\}$. Clearly $\mathfrak{N}$ is a quasimodel for $\mathcal{K}$. In particular, it is guaranteed that for all existential restrictions and context operators 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 enumerates all possible quasistates and then iteratively eliminates all those which violate any of the constraints defined above. If the elimination terminates returning a non-empty set of quasistates each containing at least one object type, then this set is guaranteed to be a quasimodel and the search is finished with the answer "$\mathcal{K}$ is satisfiable". Else, no quasimodel exists and the algorithm returns "$\mathcal{K}$ is unsatisfiable".

We start by enumerating the set $\mathfrak{N}$ of all possible quasistates. Further, we enumerate all possible mappings $\gamma : ind_c(\mathcal{K}) \mapsto \mathfrak{N}$. The algorithm proceeds in two steps:

1. select a mapping γ, and for every $\boldsymbol{c} \in ind_c(\mathcal{K})$ eliminate all quasistates $q \in \mathfrak{N}$ such that $q \neq \gamma(\boldsymbol{c})$, with $q = \langle c_q, f_q, O_q \rangle$ and $\{\boldsymbol{c}\} \in c_q$,

2. iteratively eliminate all quasistates and object types from the quasistates which violate any of the conditions (qS), (qC1)-(qC4), (QS1)-(QS3), (M1)-(M2).

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.

In such case the result of elimination is clearly a quasimodel and the search is finished with the answer "$\mathcal{K}$ is satisfiable". Else, if all quasistates get eliminated, the algorithm selects another mapping γ and repeats the elimination procedure. If none of the mappings allow for a successful termination then clearly no quasimodel exists and the algorithm returns "$\mathcal{K}$ is unsatisfiable".

The whole algorithm runs in double exponential time in the size of $\mathcal{K}$. To show this, we observe that the following (very liberally estimated) inequalities hold. By $\ell(\mathcal{K})$ we denote the size of $\mathcal{K}$, measured in the number of symbols used, and by $|X|$ — the number of elements of set X:

$$|con_c(\mathcal{K}) \cup con_o(\mathcal{K})| \leq 2\ell(\mathcal{K}),$$
$$|ind_c(\mathcal{K})| \leq \ell(\mathcal{K}), \;\; |sub_o(\mathcal{K})| \leq \ell(\mathcal{K}),$$

size of a quasistate:

$$\ell(q) \leq \ell(\mathcal{K}) \cdot (|con_c(\mathcal{K})| + |sub_o(\mathcal{K})| + 2^{|con_o(\mathcal{K})|}) \leq \ell(\mathcal{K}) \cdot (2\ell(\mathcal{K}) + \ell(\mathcal{K}) + 2^{2\ell(\mathcal{K})}),$$

number of quasistates in a quasimodel:

$$|\mathfrak{N}| = 2^{|con_c(\mathcal{K})|} \cdot 2^{|sub_o(\mathcal{K})|} \cdot 2^{2^{|con_o(\mathcal{K})|}} = 2^{2\ell(\mathcal{K})} \cdot 2^{\ell(\mathcal{K})} \cdot 2^{2^{2\ell(\mathcal{K})}}.$$

Since deciding whether a quasistate can be eliminated at a given stage, in particular checking if there exist appropriate witnesses for it (QS1)-(QS3), cannot take more than $\ell(q)^2 \cdot |\mathfrak{N}|$ steps, therefore a single run of the elimination procedure takes no more than $(\ell(q) \cdot |\mathfrak{N}|)^2$ steps. Finally, there can be at most $|\mathfrak{N}|^{|ind_c(\mathcal{K})|}$ different mappings γ, hence the whole procedure must terminate in time belonging to $O(2^{2^{\ell(\mathcal{K})}})$. ❑

Lower complexity bound

Next, we derive the lower bound for the concept satisfiability problem in the logic $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$, which carries over to several other logics discussed in this paper, including $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$. We start by making an observation, which is especially useful in the proof, 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 2 *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 $\mathfrak{M} = (\mathfrak{W}, \{<_i\}_{1\leq i\leq n}, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w\in\mathfrak{W}})$, such that $\langle\mathfrak{W}, \bigcup\{<_i\}_{1\leq i\leq n}\rangle$ is a tree, every world in $\mathfrak{W}$ 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 arbitrary $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-models by using the standard unraveling technique. Thus, in what follows, we focus exclusively on $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$*-tree-models.*

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 space-bounded *Alternating Turing Machine* (ATM), which is known to be 2ExpTime-hard [16].

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 $\omega q \omega'$, where $\omega, \omega' \in (\Gamma \setminus \{\varnothing\})^*$ and $q \in Q$, which says that the tape contains the word $\omega\omega'$ (possibly followed by blank symbols), the machine is in state q and the head of the machine is on the leftmost symbol of ω'. A succeeding configuration is defined by transitions δ, where the head of the machine reads and writes the symbols on the tape. A configuration $\omega q \omega'$ 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 [4]. 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* $q_0\omega$, where ω is of length n,

- every configuration $\omega q \omega'$ on the tree, where $\omega\omega'$ is of length at most 2^n, is succeeded 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 ω iff there exists an accepting ATM tree with $q_0\omega$ as its initial configuration. The set of all words accepted by an ATM $\mathcal{M}$ is denoted as the language $L(\mathcal{M})$. As demonstrated in [16, Theorem 3.4], the problem of deciding whether $\omega \in L(\mathcal{M})$, for ω 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 4). 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 5).

Let $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$ be an ATM and ω the word for which we want to decide whether $\omega \in L(\mathcal{M})$. In the following we will construct a TBox $\mathcal{T}_\mathcal{M}$ and a concept $C_{\mathcal{M},\omega}$, of a total polynomial size in the size of the input, such that $\omega \in L(\mathcal{M})$ *iff* $C_{\mathcal{M},\omega}$ 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):

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

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 (1)-(5) implement an exemplary *downward counter*, based on atomic concepts X_i, for $1 \leq i \leq n$, which simulate bits in a binary number (X_1 stands for the least significant bit). 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.

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

$$\neg X_i \sqcap \neg X_j \sqsubseteq \Box \neg X_i, \text{ for every } 1 \leq j < i \leq n, \tag{2}$$

$$X_i \sqcap \neg X_j \sqsubseteq \Box X_i, \text{ for every } 1 \leq j < i \leq n, \tag{3}$$

$$\neg X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \Box X_j, \text{ for every } 1 \leq j \leq n, \tag{4}$$

$$X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \Box \neg X_j, \text{ for every } 1 \leq j \leq n. \tag{5}$$

An alternative *upward counter*, initiated with $Count_u$ and implemented via template (6)-(10), behaves exactly the same way, with the only difference that the counting proceeds along the alphabet modalities *up* the branch of the model.

$$Count_u \equiv \prod_{j=1}^{n} X_j, \tag{6}$$

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

$$\Diamond(\neg X_i \sqcap X_j) \sqsubseteq \neg X_i, \text{ for every } 1 \leq j < i \leq n, \tag{8}$$

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

$$\Diamond(\neg X_j \sqcap \neg X_{j-1} \sqcap \ldots \sqcap \neg X_1) \sqsubseteq X_j, \text{ for every } 1 \leq j \leq n. \tag{10}$$

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

$$Count_d^{tape} \equiv \prod_{j=1}^{n} \neg R_j, \tag{11}$$

and define constraints which encode a single tape on a branch of a model. In (12) we define the beginning of such a tape, in (13) its end, while with (14)-(16) 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 4. In fact, in our models we will need only one such individual which will single out the whole structure of the ATM tree. Constraint (16) ensures that the blank symbol is followed only by blank symbols on the tape.

$$\mathit{StartTape} \equiv \mathit{Tape} \sqcap \mathit{Count}_d^{tape}, \tag{12}$$

$$\mathit{EndTape} \equiv \mathit{Tape} \sqcap \Diamond \mathit{Count}_d^{tape}, \tag{13}$$

$$\mathit{Tape} \sqcap \neg \mathit{EndTape} \sqsubseteq \Diamond \mathit{Tape}, \tag{14}$$

$$\Diamond(\mathit{Tape} \sqcap \neg \mathit{StartTape}) \sqsubseteq \mathit{Tape}, \tag{15}$$

$$\bigcirc_a \mathit{Tape} \sqcap \bigcirc_b \mathit{Tape} \sqsubseteq \bot, \text{ for every } a \neq b \in \Gamma, \tag{16}$$

$$\bigcirc_\varnothing (\mathit{Tape} \sqcap \bigcirc_a \mathit{Tape}) \sqsubseteq \bot, \text{ for every } a \neq \varnothing \in \Gamma. \tag{17}$$

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 5.

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} (18) 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 (19) or existential (20) and then carried to the end of the tape. There the transitions take place (21)-(22) and new tapes are initiated.

$$\mathit{Count}_d^{head} \equiv \bigsqcap_{j=1}^{n} \neg S_j, \tag{18}$$

$$\bigcirc_a(Q_q \sqcap \mathit{Tape}) \sqsubseteq \bigcirc_a(\bigsqcap_{(q'b'm)\in\delta(q,a)} M_{q',b,m} \sqcap \mathit{Count}_d^{head}), \quad \text{for every } a \in \Gamma, q \in Q_\forall, \tag{19}$$

$$\bigcirc_a(Q_q \sqcap \mathit{Tape}) \sqsubseteq \bigcirc_a(\bigsqcup_{(q'b'm)\in\delta(q,a)} M_{q',b,m} \sqcap \mathit{Count}_d^{head}), \quad \text{for every } a \in \Gamma, q \in Q_\exists, \tag{20}$$

$$M_{q,a,m} \sqsubseteq \Box M_{q,a,m}, \tag{21}$$

$$M_{q,a,m} \sqcap \mathit{EndTape} \sqsubseteq \bigcirc_{q,a,m} \Diamond \mathit{StartTape}, \text{ for every } (q,a,m) \in \Theta. \tag{22}$$

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

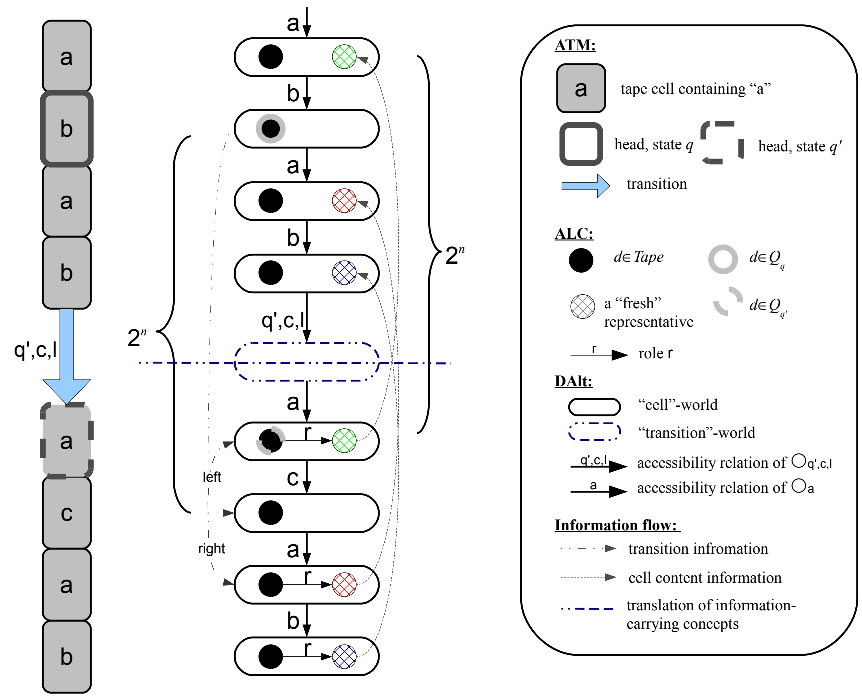

Figure 5: A transition between succeeding configurations in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models for $n = 2$ and $(q', c, l) \in \delta(q, b)$.

copies $N_{q,a,m}$ for all concepts $M_{q,a,m}$, which continue to carry their information over the new tape (23)-(24). 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 (25)-(27). Its use is necessary to avoid potential clashes between head counters initiated on two consecutive tapes. Obviously, the positions of the head over two such tapes might be separated by exactly 2^n counting steps (when the head stays at the same position), but not only: also $2^n - 1$ (whenever the head moves leftwards) or $2^n + 1$ (whenever the head moves rightwards). This observation is reflected in the constraints (28)-(30), which introduce some handy abbreviations used further for imposing the new configuration.

$$M_{q,a,m} \sqsubseteq \bigcirc_{q,a,m} N_{q,a,m}, \tag{23}$$

$$N_{q,a,m} \sqsubseteq \Box N_{q,a,m}, \tag{24}$$

$$Count_d^{*head} \equiv \prod_{j=1}^{n} \neg T_j, \tag{25}$$

$$S_i \sqsubseteq \blacksquare T_i, \text{ for every } 1 \leq i \leq n, \tag{26}$$

$$\neg S_i \sqsubseteq \blacksquare \neg T_i, \text{ for every } 1 \leq i \leq n, \tag{27}$$

$$Count_d^{*head} - 1 \equiv Head^l, \tag{28}$$

$$Count_d^{*head} \equiv Head^n, \tag{29}$$

$$Count_d^{*head} + 1 \equiv Head^r. \tag{30}$$

The necessary changes in the configuration are imposed through constraints (31)-(32), 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 (33)-(34) ensure that the transition does not push the head beyond the tape.

$$N_{q,a,m} \sqcap Tape \sqcap Head^m \sqsubseteq Q_q, \text{ for every } (q, a, m) \in \Theta, \tag{31}$$

$$\bigcirc_b \left(N_{q,a,m} \sqcap Tape \sqcap Head^n\right) \sqsubseteq \bot, \text{ for every } (q, a, m) \in \Theta \text{ and } b \neq a \in \Gamma, \tag{32}$$

$$Head^n \sqcap StartTape \sqsubseteq \neg N_{q,a,l}, \text{ for every } q \in Q, a \in \Gamma, \tag{33}$$

$$Head^n \sqcap EndTape \sqsubseteq \neg N_{q,a,r}, \text{ for every } q \in Q, a \in \Gamma. \tag{34}$$

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 (35)-(38).

$$Count_u^{cell} \equiv \prod_{j=1}^{n} U_j, \tag{35}$$

$$Count_u^{*cell} \equiv \prod_{j=1}^{n} V_j, \tag{36}$$

$$\blacklozenge U_i \sqsubseteq V_i, \text{ for every } 1 \leq i \leq n, \tag{37}$$

$$\blacklozenge \neg U_i \sqsubseteq \neg V_i, \text{ for every } 1 \leq i \leq n. \tag{38}$$

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 (39)-(40) and aligned at the transition point (41). Constraint (42) 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 (43).

$$\Diamond W_a \sqsubseteq W_a, \text{ for every } a \in \Gamma, \tag{39}$$

$$\Diamond S_a \sqsubseteq S_a, \text{ for every } a \in \Gamma, \tag{40}$$

$$\blacklozenge W_a \sqsubseteq S_a, \text{ for every } a \in \Gamma, \tag{41}$$

$$\bigcirc_a (\mathit{Tape} \sqcap \neg \mathit{Head}^n) \sqsubseteq \bigcirc_a \exists r.(\mathit{Count}_u^{cell} \sqcap W_a), \text{ for every } a \in \Gamma, \tag{42}$$

$$\bigcirc_a (S_b \sqcap \mathit{Count}_u^{*cell}) \sqsubseteq \bot, \text{ for every } b \neq a \in \Gamma. \tag{43}$$

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

$$\top \sqsubseteq \neg Q_{q_r}. \tag{44}$$

This completes the construction of the TBox $\mathcal{T}_{\mathcal{M}}$. The initial configuration $q_0\omega$ is encoded as concept $C_{\mathcal{M},\omega}$. Let $\omega = a_1 \dots a_n$. 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 $C_{\mathcal{M},\omega} = \bigcirc_{a_1}(\mathit{StartTape} \sqcap Q_{q_0} \sqcap A_2)$. We conclude by demonstrating validity of the target claim:

Lemma 2 $\omega \in L(\mathcal{M})$ iff $C_{\mathcal{M},\omega}$ *is satisfiable w.r.t. global* $\mathcal{T}_{\mathcal{M}}$ *in* $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$.

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

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

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

Then for every transition (q, a, m) from $\omega q \omega'$ in T, resulting in the succeeding configuration $\varpi q' \varpi'$, pick the world $w \in \mathfrak{W}$ such that $x <_{q,a,m} w$ and repeat the procedure above for the tape of $\varpi q' \varpi'$ starting from the world w. 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 ω from a $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-model we only need to pick an individual d, such that $d \in C_{\mathcal{M},\omega}^{\mathcal{I}(w_0)}$ and follow the paths of worlds $w \in \mathfrak{W}$ for which $d \in Tape^{\mathcal{I}(w)}$, just as presented in Figure 4. 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. ❑