Towards More Effective Tableaux Reasoning for CKR

Loris Bozzato[1], Martin Homola[2], and Luciano Serafini[1]

[1] Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy
[2] FMFI, Comenius University, Mlynská dolina, 84248 Bratislava, Slovakia

{bozzato,serafini}@fbk.eu, homola@fmph.uniba.sk

1 Introduction

Representation of context dependent knowledge in the Semantic Web is a recently emergent issue. A number of logical formalisms with this aim have been proposed [3, 11, 14]. Among them is the DL based Contextualized Knowledge Repository (CKR) [13]. One of the mostly advocated advantages of context based knowledge representation is that reasoning procedures can be constructed by composing local reasoners running inside each context, with the obvious divide-and-conquer advantage.

We recently proposed a tableaux decision algorithm [5, 9] for the case of CKR framework based on $\mathcal{ALC}$ DL. The algorithm extends the well known $\mathcal{ALC}$ tableaux algorithm [6, 12] and it is based on a combination of local reasoning inside each context with a set of novel rules that propagate knowledge across the neighboring contexts. To our best knowledge, it is the only direct tableaux reasoning algorithm for contextualized DL knowledge to date: by direct we mean not based on some reduction to a single DL knowledge base, which neglects the divide-and-conquer advantage.

In this paper, we review this algorithm and we describe our initial ideas on possible optimization, including dimensional coverage caching and parallelization. In order to maximize the divide-and-conquer advantage, it is important to propagate only those symbols between local tableaux which are really needed to assure completeness. We propose a (correctness preserving) modification of three propagation rules that decreases the amount of propagation and also of related non-deterministic branching. Proofs of all theorems can be found in the accompanying technical report [9].

2 Contextualized Knowledge Repositories

We briefly introduce the basic definition of CKR, for all details see [13]. A *meta vocabulary* Γ is used to state information about contexts. It contains contextual attributes (called dimensions), their possible values and coverage relations between these values. Formally, it is a DL vocabulary that contains: (a) a finite set of individuals called *context identifiers*; (b) a finite set of roles $\mathbf{A}$ called *dimensions*; (c) for every dimension $A \in \mathbf{A}$, a finite set of individuals D_A, called *dimensional values*, and a role $\prec_A$, called *coverage relation*. The number of dimensions $k = |\mathbf{A}|$ is assumed to be a fixed constant.

Dimensional vectors are used to identify each context with a specific set of dimensional values. Given a meta-vocabulary Γ with dimensions $\mathbf{A} = \{A_1, \ldots, A_k\}$, a *dimensional vector* $\mathbf{d} = \{A_{i_1}{:=}d_1, \ldots, A_{i_m}{:=}d_m\}$ is a (possibly empty) set of assignments such that for every j, h, with $1 \leq j \leq h \leq m$, $d_j \in D_{A_{i_j}}$, and $j \neq h$ implies $i_j \neq i_h$. A dimensional vector $\mathbf{d}$ is full if it assigns values to all dimensions (i.e., $|\mathbf{d}| = k$), otherwise it is partial. If it is apparent which value belongs to which dimension, we simply write $\{d_1, \ldots, d_m\}$. By d_A (e_A, etc.) we denote the actual value that $\mathbf{d}$ ($\mathbf{e}$, etc.) assigns to the dimension A. The *dimensional space* $\mathfrak{D}_\Gamma$ of Γ is the set of all full dimensional vectors of Γ.

An *object-vocabulary*, encodes knowledge inside contexts: it is a standard DL vocabulary Σ (with disjoint sets N_C of atomic concepts, N_R of roles and N_I of individuals) closed w.r.t. *concept/role qualification*. That is, for every concept or role symbol X of Σ and every (possibly partial) dimensional vector $\mathbf{d}$, a new symbol $X_\mathbf{d}$ is added to Σ, called the *qualification* of X w.r.t. $\mathbf{d}$. If $\mathbf{d}$ is partial then $X_\mathbf{d}$ is partially qualified, if $\mathbf{d}$ is full, it is fully qualified. Qualified symbols are used inside contexts to refer to the meaning of symbols w.r.t. some other context. This will become apparent from the semantics. Contexts and CKR knowledge bases are formally defined as follows.

Definition 1 (Context). *Given a pair of meta and object vocabularies $\langle \Gamma, \Sigma \rangle$, a* context *is a triple $\langle \mathcal{C}, \dim(\mathcal{C}), \mathrm{K}(\mathcal{C}) \rangle$ where: $\mathcal{C}$ is a context identifier of Γ; $\dim(\mathcal{C})$ is a full dimensional vector of $\mathfrak{D}_\Gamma$; and $\mathrm{K}(\mathcal{C})$ is an $\mathcal{ALC}$ knowledge base over Σ.*

Definition 2 (Contextualized Knowledge Repository). *Given a pair of meta and object vocabularies $\langle \Gamma, \Sigma \rangle$, a* CKR knowledge base (CKR) *is a pair $\mathfrak{K} = \langle \mathfrak{M}, \mathfrak{C} \rangle$ where $\mathfrak{C}$ is a set of contexts on $\langle \Gamma, \Sigma \rangle$ and $\mathfrak{M}$, called* meta knowledge*, is a DL knowledge base over Γ such that:*

(a) for $A \in \mathbf{A}$ and $d, d' \in D_A$, if $\mathfrak{M} \models A(\mathcal{C}, d)$ and $\mathfrak{M} \models A(\mathcal{C}, d')$ then $\mathfrak{M} \models d = d'$;
(b) for $\mathcal{C} \in \mathfrak{C}$ with $\dim(\mathcal{C}) = \mathbf{d}$ and for $A \in \mathbf{A}$, we have $\mathfrak{M} \models A(\mathcal{C}, d_A)$;
(c) the relation $\{\langle d, d' \rangle \mid \mathfrak{M} \models d \prec_A d'\}$ is a strict partial order on D_A.

In the rest of the paper we assume that CKR knowledge bases are defined over some suitable vocabulary $\langle \Gamma, \Sigma \rangle$, and all concepts are in negation normal form (NNF, see [1]). We also assume the unique name assumption (UNA) for the meta knowledge (i.e., if $a \neq b$ are two different symbols then $\mathfrak{M} \not\models a = b$). This is just to avoid the confusing possibility of two contexts located as the same place in the dimensional space.

For a CKR $\mathfrak{K}$, we will denote by $\mathcal{C}_\mathbf{d}$ a context with $\dim(\mathcal{C}) = \mathbf{d}$. For $\mathbf{d}, \mathbf{e} \in \mathfrak{D}_\Gamma$ and $\mathbf{B}, \mathbf{C} \subseteq \mathbf{A}$, $\mathbf{d}_\mathbf{B} := \{(A{:=}d) \in \mathbf{d} \mid A \in \mathbf{B}\}$ is the projection of $\mathbf{d}$ w.r.t. $\mathbf{B}$; and $\mathbf{d}_\mathbf{B} + \mathbf{e}_\mathbf{C} := \mathbf{d}_\mathbf{B} \cup \{(A{:=}d) \in \mathbf{e}_\mathbf{C} \mid A \notin \mathbf{B}\}$ is the completion of $\mathbf{d}_\mathbf{B}$ w.r.t. $\mathbf{e}_\mathbf{C}$.

An important notion is the strict ($\prec$) and non-strict ($\preceq$) coverage between dimensional values: for $d, d' \in D_A$, $d \prec d'$ if $\mathfrak{M} \models d \prec_A d'$; and $d \preceq d'$ if either $d \prec d'$ or $\mathfrak{M} \models d = d'$. Similarly, coverage for dimensional vectors: $\mathbf{d} \preceq_\mathbf{B} \mathbf{e}$ for some $\mathbf{B} \subseteq \mathbf{A}$ if $d_B \preceq e_B$ for each $B \in \mathbf{B}$; and $\mathbf{d} \prec_\mathbf{B} \mathbf{e}$ if $\mathbf{d} \preceq_\mathbf{B} \mathbf{e}$ and $d_B \prec e_B$ for at least one $B \in \mathbf{B}$. Also, $\mathbf{d} \preceq \mathbf{e}$ if $\mathbf{d} \preceq_\mathbf{A} \mathbf{e}$, and $\mathbf{d} \prec \mathbf{e}$ if $\mathbf{d} \prec_\mathbf{A} \mathbf{e}$. Finally coverage for contexts: $\mathcal{C}_\mathbf{d} \preceq \mathcal{C}_\mathbf{e}$ if $\mathbf{d} \preceq \mathbf{e}$, and $\mathcal{C}_\mathbf{d} \prec \mathcal{C}_\mathbf{e}$ if $\mathbf{d} \prec \mathbf{e}$. Intuitively, if $\mathcal{C}_\mathbf{d} \prec \mathcal{C}_\mathbf{e}$, then $\mathcal{C}_\mathbf{d}$ is the narrower and $\mathcal{C}_\mathbf{e}$ is the broader context.

An example CKR $\mathfrak{K}_{fb}$ shown in Fig. 1 uses three dimensions time, $\mathsf{location}$, and topic. It has four contexts associated with dimensional vectors $\mathbf{sp}$ (general context of sports in 2010), $\mathbf{fb}$ (football in 2010), $\mathbf{wc10}$ (FIFA World Cup 2010), and $\mathbf{nfl10}$

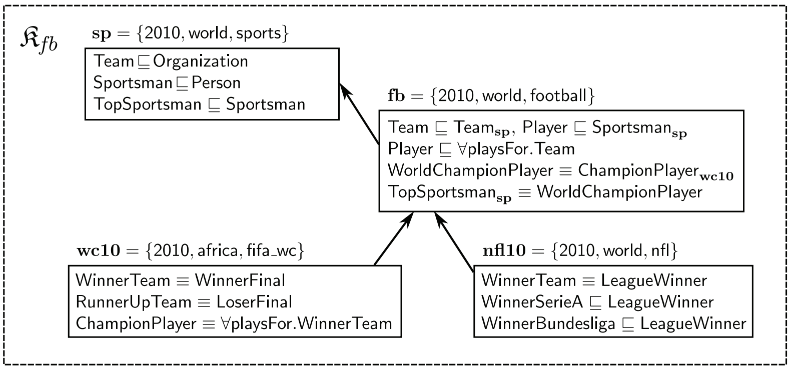

Fig. 1. Example CKR knowledge base $\mathfrak{K}_{fb}$

(national football leagues in 2010). Axioms are placed inside each context while the associated vector is placed above it. Coverage relation $\prec$ is visualized with arrows.

Note that in CKR built on top of more expressive logics, conditions 2 (a,c) of Definition 2 can be assured directly in the meta knowledge with respective axioms: each $A \in \mathbf{A}$ is declared functional, and each $\prec_A$ is declared irreflexive and transitive. In $\mathcal{ALC}$ we do not have this option, however this is not a problem, because the number of all dimensions is assumed to be finite as it is the number of contexts in a CKR. Hence after the meta knowledge is modeled, these conditions can be verified even without a reasoner (e.g., by some script). These conditions are needed to assure reasonable properties of contextual space, i.e., acyclicity, dimensional values uniquely determined [13].

CKR uses DL semantics inside each context combined with some additional semantic restrictions to ensure proper meaning of qualified symbols. A *partial DL interpretation* of a DL vocabulary Σ is a DL interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ that allows two exceptions: $\Delta^{\mathcal{I}}$ is possibly an empty set, and $\cdot^{\mathcal{I}}$ is totally defined on N_{C} and N_{R} and it is partially defined on N_{I} (i.e., $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ can be undefined for some $a \in N_{\mathrm{I}}$). Partial interpretations need not necessarily provide denotations for all individuals of Σ. This is needed for technical reasons: intuitively, all contexts rely on the same object vocabulary Σ, but some element of Σ may not be meaningful in all contexts. Also, interpretations with empty domains are useful to treat inconsistency among contexts [13].

Definition 3 (CKR-Model). *A model of a CKR $\mathfrak{K}$ is a collection $\mathfrak{I} = \{\mathcal{I}_{\mathbf{d}}\}_{\mathbf{d} \in \mathfrak{D}_\Gamma}$ of partial DL interpretations (*local interpretations*) s.t. for all $\mathbf{d}, \mathbf{e}, \mathbf{f} \in \mathfrak{D}_\Gamma$, $\mathbf{B} \subseteq \mathbf{A}$, $A \in N_{\mathrm{C}}$, $R \in N_{\mathrm{R}}$, $X \in N_{\mathrm{C}} \cup N_{\mathrm{R}}$, $a \in N_{\mathrm{I}}$:*

1. $(\top_{\mathbf{d}})^{\mathcal{I}_{\mathbf{f}}} \subseteq (\top_{\mathbf{e}})^{\mathcal{I}_{\mathbf{f}}}$ *if* $\mathbf{d} \prec \mathbf{e}$
2. $(A_{\mathbf{f}})^{\mathcal{I}_{\mathbf{d}}} \subseteq (\top_{\mathbf{f}})^{\mathcal{I}_{\mathbf{d}}}$
3. $(R_{\mathbf{f}})^{\mathcal{I}_{\mathbf{d}}} \subseteq (\top_{\mathbf{f}})^{\mathcal{I}_{\mathbf{d}}} \times (\top_{\mathbf{f}})^{\mathcal{I}_{\mathbf{d}}}$
4. $a^{\mathcal{I}_{\mathbf{e}}} = a^{\mathcal{I}_{\mathbf{d}}}$ *if* $\mathbf{d} \prec \mathbf{e}$ *and*
   - $a^{\mathcal{I}_{\mathbf{d}}}$ *is defined or,*
   - $a^{\mathcal{I}_{\mathbf{e}}}$ *is defined and* $a^{\mathcal{I}_{\mathbf{e}}} \in \Delta_{\mathbf{d}}$
5. $(X_{\mathbf{d}_{\mathbf{B}}})^{\mathcal{I}_{\mathbf{e}}} = (X_{\mathbf{d}_{\mathbf{B}}+\mathbf{e}})^{\mathcal{I}_{\mathbf{e}}}$
6. $(X_{\mathbf{d}})^{\mathcal{I}_{\mathbf{e}}} = (X_{\mathbf{d}})^{\mathcal{I}_{\mathbf{d}}}$ *if* $\mathbf{d} \prec \mathbf{e}$
7. $(A_{\mathbf{f}})^{\mathcal{I}_{\mathbf{d}}} = (A_{\mathbf{f}})^{\mathcal{I}_{\mathbf{e}}} \cap \Delta_{\mathbf{d}}$ *if* $\mathbf{d} \prec \mathbf{e}$
8. $(R_{\mathbf{f}})^{\mathcal{I}_{\mathbf{d}}} = (R_{\mathbf{f}})^{\mathcal{I}_{\mathbf{e}}} \cap (\Delta_{\mathbf{d}} \times \Delta_{\mathbf{d}})$ *if* $\mathbf{d} \prec \mathbf{e}$
9. $\mathcal{I}_{\mathbf{d}} \models \mathrm{K}(\mathcal{C}_{\mathbf{d}})$

The semantics takes care that local domains respect the coverage hierarchy (condition 1). Note that $\top_{\mathbf{d}}$ represents the domain of $\mathcal{I}_{\mathbf{d}}$ in the context where it appears. It gives rigid meaning to individuals, however, the meaning of an individual in a super-context is independent if its meaning in a sub-context is undefined (condition 4). The interpretation of any $X_{\mathbf{f}}$ in any context $\mathcal{C}_{\mathbf{d}}$ is roofed under $(\top_{\mathbf{f}})^{\mathcal{I}_{\mathbf{d}}}$ (conditions 2, 3). The meaning of $X_{\mathbf{f}}$ in some context $\mathcal{C}_{\mathbf{e}}$ is based on its context of origin $\mathcal{C}_{\mathbf{f}}$ if this context is less specific than $\mathcal{C}_{\mathbf{e}}$ (condition 6); otherwise, at least, any $X_{\mathbf{f}}$ in $\mathcal{C}_{\mathbf{d}}$ and $\mathcal{C}_{\mathbf{e}}$ must be equal on the shared part of their domains (conditions 7 and 8). Finally, each $\mathcal{I}_{\mathbf{d}}$ is a DL-model of $\mathcal{C}_{\mathbf{d}}$ (condition 9). Albeit useful for modeling, partially qualified symbols are a kind of syntactic sugar in this framework as the completed version of the symbol can always be used instead (condition 5, cf. [13]). To simplify the algorithm, we assume w.l.o.g. that the CKR on the input is always fully qualified. In the examples we use non-qualified symbols only for improving readability.

Given a CKR $\mathfrak{K}$ and $\mathbf{d} \in \mathfrak{D}_{\Gamma}$, a concept C is $\mathbf{d}$-*satisfiable* w.r.t. $\mathfrak{K}$ if there exists a CKR model $\mathfrak{I} = \{\mathcal{I}_{\mathbf{e}}\}_{\mathbf{e}\in\mathfrak{D}_{\Gamma}}$ of $\mathfrak{K}$ such that $C^{\mathcal{I}_{\mathbf{d}}} \neq \emptyset$; $\mathfrak{K}$ is $\mathbf{d}$-satisfiable if it has a CKR model $\mathfrak{I} = \{\mathcal{I}_{\mathbf{e}}\}_{\mathbf{e}\in\mathfrak{D}_{\Gamma}}$ such that $\Delta_{\mathbf{d}} \neq \emptyset$; $\mathfrak{K}$ is *globally satisfiable* if it has a CKR model $\mathfrak{I} = \{\mathcal{I}_{\mathbf{e}}\}_{\mathbf{e}\in\mathfrak{D}_{\Gamma}}$ such that $\Delta_{\mathbf{e}} \neq \emptyset$ for every $\mathbf{e} \in \mathfrak{D}_{\Gamma}$. An axiom α is $\mathbf{d}$-*entailed* by $\mathfrak{K}$ (denoted $\mathfrak{K} \models \mathbf{d} : \alpha$) if for every model $\mathfrak{I} = \{\mathcal{I}_{\mathbf{e}}\}_{\mathbf{e}\in\mathfrak{D}_{\Gamma}}$ of $\mathfrak{K}$ it holds $\mathcal{I}_{\mathbf{d}} \models \alpha$. As usual, $\mathbf{d}$-entailment can be reduced to $\mathbf{d}$-satisfiability: in particular $\mathfrak{K} \models \mathbf{d} : C \sqsubseteq D$ iff $C \sqcap \neg D$ is not $\mathbf{d}$-satisfiable w.r.t. $\mathfrak{K}$.

3 Tableaux Algorithm for CKR

We denote by $\mathrm{clos}(C)$ the set of all syntactically correct atomic and complex concepts that occur in a concept C. The closure of a concept C w.r.t. a CKR $\mathfrak{K}$ is $\mathrm{clos}_{\mathfrak{K}}(C) = \mathrm{clos}(C) \cup \{\mathrm{clos}(\neg D \sqcup E) \mid D \sqsubseteq E \in \mathrm{K}(\mathcal{C})$ for some context $\mathcal{C}$ of $\mathfrak{K}\} \cup \{\mathrm{clos}(D) \mid D(a) \in \mathrm{K}(\mathcal{C})$ for some context $\mathcal{C}$ of $\mathfrak{K}\} \cup \{\mathrm{clos}(\neg\top_{\mathbf{e}} \sqcup \top_{\mathbf{f}}) \mid \mathbf{e} \prec \mathbf{f}\}$. We denote with $\mathcal{R}_{\mathfrak{K},C}$ the set of roles $R \in N_{\mathrm{R}}$ appearing in C or some $\mathrm{K}(\mathcal{C})$ of $\mathfrak{K}$. The sets $\mathrm{clos}_{\mathfrak{K}}(C)$ and $\mathcal{R}_{\mathfrak{K},C}$ contain all possible concepts and roles relevant in order to verify $\mathbf{d}$-satisfiability of C w.r.t. $\mathfrak{K}$.

The tableaux algorithm $C_{\mathcal{T}}$ for CKR decides the $\mathbf{d}$-satisfiability of a concept C w.r.t. a CKR $\mathfrak{K}$: it is partly based on the well known $\mathcal{ALC}$ tableaux algorithm [12, 6] which is extended in order to deal with multiple contexts. The algorithm works on a *completion tree*, a partial representation of a CKR model that the algorithm incrementally builds.

Definition 4 (Completion tree). *Given a CKR $\mathfrak{K}$, a completion tree is a triple $T = \langle V, E, \mathcal{L}\rangle$ s.t.:*

1. *$\langle V, E\rangle$ is a tree, where V is an ordered set of elements with order $<_V$;*
2. *there is a collection $\{V_{\mathbf{d}}\}_{\mathbf{d}\in\mathfrak{D}_{\Gamma}}$ of sets such that $V_{\mathbf{d}} \subseteq V$;*
3. *$E_{\mathbf{d}} = \{\langle x, y\rangle \in E \mid x, y \in V_{\mathbf{d}}\}$, for each $\mathbf{d} \in \mathfrak{D}_{\Gamma}$;*
4. *$\mathcal{L} = \{\mathcal{L}_{\mathbf{d}}\}_{\mathbf{d}\in\mathfrak{D}_{\Gamma}}$ is a collection of labeling functions such that for each $\mathbf{d} \in \mathfrak{D}_{\Gamma}$:*
   (a) $\mathcal{L}_{\mathbf{d}}(x) \subseteq \mathrm{clos}_{\mathfrak{K}}(C)$, for each $x \in V_{\mathbf{d}}$;
   (b) $\mathcal{L}_{\mathbf{d}}(\langle x, y\rangle) \subseteq \mathcal{R}_{\mathfrak{K},C}$, for each $\langle x, y\rangle \in E_{\mathbf{d}}$.

In order to verify $\mathbf{d}$-satisfiability of a concept C w.r.t. a CKR $\mathfrak{K}$, the algorithm initializes and then iteratively expands the tree using a number of tableaux expansion rules. To avoid infinite looping, a blocking policy adapted from Buchheit et al. [6] is used. We assume that the algorithm always adds nodes into the completion tree respecting the order $<_V$ (i.e., whenever a new node x is added, $y <_V x$ holds for all y already in V).

Table 1. CKR completion rules

$\sqcap$-rule:	**if** $x \in V_{\mathbf{d}}$, $C_1 \sqcap C_2 \in \mathcal{L}_{\mathbf{d}}(x)$, $\{C_1, C_2\} \not\subseteq \mathcal{L}_{\mathbf{d}}(x)$ **then** $\mathcal{L}_{\mathbf{d}}(x) := \mathcal{L}_{\mathbf{d}}(x) \cup \{C_1, C_2\}$
$\sqcup$-rule:	**if** $x \in V_{\mathbf{d}}$, $C_1 \sqcup C_2 \in \mathcal{L}_{\mathbf{d}}(x)$, $\{C_1, C_2\} \cap \mathcal{L}_{\mathbf{d}}(x) = \emptyset$ **then** $\mathcal{L}_{\mathbf{d}}(x) := \mathcal{L}_{\mathbf{d}}(x) \cup \{C_1\}$ or $\mathcal{L}_{\mathbf{d}}(x) := \mathcal{L}_{\mathbf{d}}(x) \cup \{C_2\}$
$\exists$-rule:	**if** $x \in V_{\mathbf{d}}$, $\exists R.C \in \mathcal{L}_{\mathbf{d}}(x)$, and there is no R-successor $y \in V_{\mathbf{d}}$ of x s.t. $C \in \mathcal{L}_{\mathbf{d}}(y)$ **then** $V_{\mathbf{d}} := V_{\mathbf{d}} \cup \{z\}$ with z new, $E_{\mathbf{d}} := E_{\mathbf{d}} \cup \{\langle x, z\rangle\}$ $\mathcal{L}_{\mathbf{d}}(\langle x, z\rangle) := \{R\}$, $\mathcal{L}_{\mathbf{d}}(z) := \{C\}$
$\forall$-rule:	**if** $x \in V_{\mathbf{d}}$, $\forall R.C \in \mathcal{L}_{\mathbf{d}}(x)$, and there exists R-successor $y \in V_{\mathbf{d}}$ of x s.t. $C \notin \mathcal{L}_{\mathbf{d}}(y)$ **then** $\mathcal{L}_{\mathbf{d}}(y) := \mathcal{L}_{\mathbf{d}}(y) \cup \{C\}$
$\mathcal{T}$-rule:	**if** $x \in V_{\mathbf{d}}$, $C \sqsubseteq D \in \mathrm{K}(\mathcal{C}_{\mathbf{d}})$, $\mathrm{nnf}(\neg C \sqcup D) \notin \mathcal{L}_{\mathbf{d}}(x)$ **then** $\mathcal{L}_{\mathbf{d}}(x) := \mathcal{L}_{\mathbf{d}}(x) \cup \{\mathrm{nnf}(\neg C \sqcup D)\}$
$\Delta\!\uparrow$-rule:	**if** $x \in V_{\mathbf{d}}$, $\mathbf{d} \prec \mathbf{e}$, $x \notin V_{\mathbf{e}}$ **then** $V_{\mathbf{e}} := V_{\mathbf{e}} \cup \{x\}$
$\Delta\!\downarrow$-rule:	**if** $x \in V_{\mathbf{e}}$, $\mathbf{d} \prec \mathbf{e}$, $\top_{\mathbf{d}} \in \mathcal{L}_{\mathbf{e}}(x)$, $x \notin V_{\mathbf{d}}$ **then** $V_{\mathbf{d}} = V_{\mathbf{d}} \cup \{x\}$
A-rule:	**if** $x \in V_{\mathbf{d}} \cap V_{\mathbf{e}}$, $\mathbf{d} \prec \mathbf{e}$ or $\mathbf{d} \succ \mathbf{e}$, $A_{\mathbf{f}} \in \mathcal{L}_{\mathbf{d}}(x)$, $A_{\mathbf{f}} \notin \mathcal{L}_{\mathbf{e}}(x)$ **then** $\mathcal{L}_{\mathbf{e}}(x) := \mathcal{L}_{\mathbf{e}}(x) \cup \{A_{\mathbf{f}}\}$
R-rule:	**if** $x, y \in V_{\mathbf{d}} \cap V_{\mathbf{e}}$, $\langle x, y\rangle \in E$, $\mathbf{d} \prec \mathbf{e}$ or $\mathbf{d} \succ \mathbf{e}$, $R_{\mathbf{f}} \in \mathcal{L}_{\mathbf{d}}(\langle x, y\rangle)$, $R_{\mathbf{f}} \notin \mathcal{L}_{\mathbf{e}}(\langle x, y\rangle)$ **then** $\mathcal{L}_{\mathbf{e}}(\langle x, y\rangle) := \mathcal{L}_{\mathbf{e}}(\langle x, y\rangle) \cup \{R_{\mathbf{f}}\}$
$\top_A$-rule:	**if** $x \in V_{\mathbf{e}}$, $A_{\mathbf{d}} \in \mathcal{L}_{\mathbf{e}}(x)$, $\top_{\mathbf{d}} \notin \mathcal{L}_{\mathbf{e}}(x)$ **then** $\mathcal{L}_{\mathbf{e}}(x) := \mathcal{L}_{\mathbf{e}}(x) \cup \{\top_{\mathbf{d}}\}$
$\top_R$-rule:	**if** $x, y \in V_{\mathbf{e}}$, $\langle x, y\rangle \in E$, $R_{\mathbf{d}} \in \mathcal{L}_{\mathbf{e}}(\langle x, y\rangle)$, $\top_{\mathbf{d}} \notin \mathcal{L}_{\mathbf{e}}(x) \cap \mathcal{L}_{\mathbf{e}}(y)$ **then** $\mathcal{L}_{\mathbf{e}}(x) := \mathcal{L}_{\mathbf{e}}(x) \cup \{\top_{\mathbf{d}}\}$, $\mathcal{L}_{\mathbf{e}}(y) := \mathcal{L}_{\mathbf{e}}(y) \cup \{\top_{\mathbf{d}}\}$
$\top_{\sqsubseteq}$-rule:	**if** $x \in V_{\mathbf{d}}$, $\mathbf{e} \prec \mathbf{f}$, $\neg\top_{\mathbf{e}} \sqcup \top_{\mathbf{f}} \notin \mathcal{L}_{\mathbf{d}}(x)$ **then** $\mathcal{L}_{\mathbf{d}}(x) := \mathcal{L}_{\mathbf{d}}(x) \cup \{\neg\top_{\mathbf{e}} \sqcup \top_{\mathbf{f}}\}$
M-rule:	**if** $a^{\mathbf{g}} \in V_{\mathbf{d}}$, $a^{\mathbf{h}} \in V_{\mathbf{e}}$, and $\mathbf{d} \preceq \mathbf{e}$, **then** $\mathrm{merge}(a^{\mathbf{g}}, a^{\mathbf{h}})$

Definition 5 (Blocking). *Given a CKR $\mathfrak{K}$ and a completion tree $T = \langle V, E, \mathcal{L}\rangle$, we say that a node $w \in V$ is the witness for $x \in V$, if $\mathcal{L}_{\mathbf{d}}(x) = \mathcal{L}_{\mathbf{d}}(w)$ for all $\mathbf{d} \in \mathfrak{D}_{\Gamma}$, $w <_V x$ and there is no $y \in V$ such that $y <_V w$ and $\mathcal{L}_{\mathbf{d}}(x) = \mathcal{L}_{\mathbf{d}}(y)$ for all $\mathbf{d} \in \mathfrak{D}_{\Gamma}$. We say that $x \in V$ is blocked by $w \in V$ if w is the witness for x.*

We say that a tableaux rule is *applicable* if all of its preconditions (the if-part of the rule) are satisfied for some node $x \in V$ or a pair of nodes $x, y \in V$ and the nodes are not blocked. A completion tree T is *complete*, if none of the tableaux rules is applicable. A completion tree $T = \langle V, E, \mathcal{L}\rangle$ contains a *clash* in a node $x \in V$, if for some $\mathbf{d} \in \mathfrak{D}_{\Gamma}$ and some concept C both $C \in \mathcal{L}_{\mathbf{d}}(x)$ and $\neg C \in \mathcal{L}_{\mathbf{d}}(x)$, or if $\bot \in \mathcal{L}_{\mathbf{d}}(x)$. We say that T is *clash-free* if no clash occurs in any of its nodes.

In initialization, ABox axioms are encoded in the initial completion tree. This technique is well known for logics like $\mathcal{ALC}$ [1]. However, we must consider that in CKR same individuals appearing in different contexts may possibly have different meanings. In the completion tree, individuals will be represented by elements of the form $a^{\mathbf{g}}$ where $a \in N_{\mathrm{I}}$ and $\mathbf{g} \in \mathfrak{D}_{\Gamma}$ identifies the context in which the individual was first introduced. To implement condition 4 of CKR-models we will merge nodes when needed.

Definition 6 (Merging). *Executing* $\mathrm{merge}(x, y)$ *on a completion tree $T = \langle V, E, \mathcal{L}\rangle$, with $x, y \in V$, transforms T as follows: a) node x is added into $V_{\mathbf{e}}$ for all $\mathbf{e} \in \mathfrak{D}_{\Gamma}$ s.t. $y \in V_{\mathbf{e}}$; b) all concepts from $\mathcal{L}_{\mathbf{e}}(y)$ are added into $\mathcal{L}_{\mathbf{e}}(x)$, for all $\mathbf{e} \in \mathfrak{D}_{\Gamma}$; c) all edges directed into/from y are redirected into/from x; d) node y is removed from V.*

Finally, the algorithm is formally defined as follows:

Definition 7 (Algorithm $C_{\mathcal{T}}$). *Given as input a CKR $\mathfrak{K}$, $\mathbf{d} \in \mathfrak{D}_{\Gamma}$, and a concept C in NNF, the algorithm $C_{\mathcal{T}}$ verifies the $\mathbf{d}$-satisfiability of C w.r.t. $\mathfrak{K}$ in the following steps:*

1. *for all* $\mathbf{e} \in \mathfrak{D}_\Gamma$*, initialize* $V_\mathbf{e}$*,* E*, and* $\mathcal{L}_\mathbf{e}$ *as follows:*
   (a) $V_\mathbf{e} := \{a^\mathbf{e} \mid C(a) \in \mathrm{K}(\mathcal{C}_\mathbf{e})\} \cup \{a^\mathbf{e}, b^\mathbf{e} \mid R(a,b) \in \mathrm{K}(\mathcal{C}_\mathbf{e})\}$*;*
   $E := \{\langle a^\mathbf{e}, b^\mathbf{e}\rangle \mid R(a,b) \in \mathrm{K}(\mathcal{C}_\mathbf{e}), \mathbf{e} \in \mathfrak{D}_\Gamma\}$*;*
   $\mathcal{L}_\mathbf{e}(a^\mathbf{e}) := \{C \mid C(a) \in \mathrm{K}(\mathcal{C}_\mathbf{e})\}$*;* $\mathcal{L}_\mathbf{e}(\langle a^\mathbf{e}, b^\mathbf{e}\rangle) := \{R \mid R(a,b) \in \mathrm{K}(\mathcal{C}_\mathbf{e})\}$*;*
   (b) $V_\mathbf{d} := V_\mathbf{d} \cup \{s_0\}$*, where* s_0 *is a new constant in* $V_\mathbf{d}$*;* $\mathcal{L}_\mathbf{d}(s_0) := \{C\}$*;*
2. *exhaustively apply completion rules of Table 1 on* T*;*
3. *once* T *is complete, answer "*C *is* $\mathbf{d}$*-satisfiable w.r.t.* $\mathfrak{K}$*" if* T *is clash-free; answer "*C *is not* $\mathbf{d}$*-satisfiable w.r.t.* $\mathfrak{K}$*" otherwise.*

The first five rules used by the algorithm (from $\sqcap$- to $\mathcal{T}$-rule) are the usual $\mathcal{ALC}$ tableaux rules [1] responsible for local reasoning inside each context. The additional rules are new and they handle propagation of information between contexts.

The $\Delta\!\uparrow$- and $\Delta\!\downarrow$-rules are responsible for propagation of nodes: if $\mathbf{d} \prec \mathbf{e}$, all nodes from $V_\mathbf{d}$ are propagated to $V_\mathbf{e}$ ($\Delta\!\uparrow$-rule), but only the nodes belonging to $\top_\mathbf{d}$ are propagated from $V_\mathbf{e}$ to $V_\mathbf{d}$ ($\Delta\!\downarrow$-rule).

Given contexts $\mathcal{C}_\mathbf{d}$ and $\mathcal{C}_\mathbf{e}$, with $\mathbf{d} \prec \mathbf{e}$, the conditions 6 and 7 of CKR-models require that the interpretations of any symbol $X_\mathbf{f}$ in the contexts agree on all elements shared by their domains. Hence, if a node (or a pair of nodes) belongs to both $V_\mathbf{d}$ and $V_\mathbf{e}$ (i.e. it belongs to both local tableaux), then its labels are propagated by A-rule and R-rule from one local tableaux to another, in both directions.

The following rules maintain the first three semantic conditions of CKR-models. The $\top_A$- and $\top_R$-rules take care that any qualified symbol $X_\mathbf{d}$ is always roofed under $\top_\mathbf{d}$ in any context $\mathcal{C}_\mathbf{e}$. If a qualified concept $A_\mathbf{d}$ (role $R_\mathbf{d}$) is found in the $\mathcal{L}_\mathbf{e}$-label of some node (edge) in $V_\mathbf{e}$, then $\top_\mathbf{d}$ is added to the $\mathcal{L}_\mathbf{e}$-label of this node (or both nodes connected by this edge). Also, if $\mathbf{e} \prec \mathbf{f}$, then the $\top_\sqsubseteq$-rule assures that the subsumption $\top_\mathbf{e} \sqsubseteq \top_\mathbf{f}$ must hold in any context. Finally, the M-rule takes care of cases when it is inferred that the same individual a appears in two different contexts.

It is however not the case that there is one-to-one correspondence between the semantic conditions of CKR (Definition 3) and the tableaux rules. Consider condition 6 and the case when $X_\mathbf{d} = A_\mathbf{d}$ and $\mathbf{d} \prec \mathbf{e}$. If for instance due to a firing of the $\exists$-rule a new node x was added into $V_\mathbf{e}$ with $\mathcal{L}_\mathbf{e}(x)$ initiated to $\{A_\mathbf{d}\}$, to maintain condition 6 the same node with the same label must also be added to $V_\mathbf{d}$ and $\mathcal{L}_\mathbf{d}$ respectively. This is achieved by consecutive firing of $\top_A$-, $\Delta\!\downarrow$-, and A-rules. A more complex example of reasoning with CKR tableaux rules follows.

Example 1 (Tableaux algorithm). Using the algorithm and our example CKR $\mathfrak{K}_{fb}$, let us show the proof for the following subsumption:

$$\mathfrak{K}_{fb} \models \mathbf{nfl10} : \mathsf{WorldChampionPlayer}_\mathbf{fb} \sqsubseteq \forall \mathsf{playsFor}_\mathbf{wc10}.\mathsf{WinnerTeam}_\mathbf{wc10}$$

Initialization yields $V_\mathbf{nfl10} := \{s_0\}$ and $\mathcal{L}_\mathbf{nfl10}(s_0) := \{\mathsf{WorldChampionPlayer}_\mathbf{fb} \sqcap \exists \mathsf{playsFor}_\mathbf{wc10}.\ \neg \mathsf{WinnerTeam}_\mathbf{wc10}\}$. Then tableaux rules are applied as follows:

(1) $\mathcal{L}_\mathbf{nfl10}(s_0) := \mathcal{L}_\mathbf{nfl10}(s_0) \cup \{\mathsf{WorldChampionPlayer}_\mathbf{fb}, \exists \mathsf{playsFor}_\mathbf{wc10}.\neg \mathsf{WinnerTeam}_\mathbf{wc10}\}$ by $\sqcap$-rule;
(2) $V_\mathbf{nfl10} := V_\mathbf{nfl10} \cup \{s_1\}$, $E_\mathbf{nfl10} := \{\langle s_0, s_1\rangle\}$, $\mathcal{L}_\mathbf{nfl10}(\langle s_0, s_1\rangle) := \{\mathsf{playsFor}_\mathbf{wc10}\}$, $\mathcal{L}_\mathbf{nfl10}(s_1) := \{\neg \mathsf{WinnerTeam}_\mathbf{wc10}\}$ by $\exists$-rule;
(3) $V_\mathbf{fb} := \{s_0, s_1\}$, $\mathcal{L}_\mathbf{fb}(s_0) := \{\mathsf{WorldChampionPlayer}\}$, $\mathcal{L}_\mathbf{fb}(\langle s_0, s_1\rangle) := \{\mathsf{playsFor}_\mathbf{wc10}\}$ by $\Delta\!\uparrow$-, A- and R-rules;

(4) $\mathcal{L}_{\mathbf{fb}}(s_0) \cup \{\mathsf{ChampionPlayer}_{\mathbf{wc10}}\}$ by $\mathcal{T}$- and $\sqcup$-rules;
(5) $\mathcal{L}_{\mathbf{fb}}(s_0) := \mathcal{L}_{\mathbf{fb}}(s_0) \cup \{\top_{\mathbf{wc10}}\}$, $\mathcal{L}_{\mathbf{fb}}(s_1) := \mathcal{L}_{\mathbf{fb}}(s_1) \cup \{\top_{\mathbf{wc10}}\}$ by $\top_R$-rule;
(6) $V_{\mathbf{wc10}} := \{s_0, s_1\}$, $\mathcal{L}_{\mathbf{wc10}}(s_0) := \{\mathsf{ChampionPlayer}\}$,
$\mathcal{L}_{\mathbf{wc10}}(\langle s_0, s_1\rangle) := \{\mathsf{playsFor}\}$ by $\Delta\!\downarrow$-, A- and R-rules;
(7) $\mathcal{L}_{\mathbf{wc10}}(s_0) := \mathcal{L}_{\mathbf{wc10}}(s_0) \cup \{\forall \mathsf{playsFor}.\mathsf{WinnerTeam}\}$ by $\mathcal{T}$- and $\sqcup$-rules;
(8) $\mathcal{L}_{\mathbf{wc10}}(s_1) := \mathcal{L}_{\mathbf{wc10}}(s_1) \cup \{\mathsf{WinnerTeam}\}$ by $\forall$-rule;
(9) $\mathcal{L}_{\mathbf{fb}}(s_1) := \mathcal{L}_{\mathbf{fb}}(s_1) \cup \{\mathsf{WinnerTeam}_{\mathbf{wc10}}\}$,
$\mathcal{L}_{\mathbf{nfl10}}(s_1) := \mathcal{L}_{\mathbf{nfl10}}(s_1) \cup \{\mathsf{WinnerTeam}_{\mathbf{wc10}}\}$ by A-rule;

The application of last rule creates a clash, since $\mathcal{L}_{\mathbf{nfl10}}(s_1) = \{\neg\mathsf{WinnerTeam}_{\mathbf{wc10}}, \mathsf{WinnerTeam}_{\mathbf{wc10}}\}$. Note that in the non-deterministic choices asked in steps 4 and 7 (due to $\sqcup$-rule), all other choices immediately lead to a clash. Hence no clash-free completion tree can be constructed and the algorithm answers that the input concept is $\mathbf{nfl10}$-unsatisfiable w.r.t. $\mathfrak{K}_{fb}$. This implies that the subsumption in question is entailed.

Note the required inter-contextual knowledge propagation: we first had to propagate nodes and their labels from $V_{\mathbf{nfl10}}$ to $V_{\mathbf{fb}}$ and finally to $V_{\mathbf{wc10}}$ by tracking the context coverage structure (steps 3–6). Then with the last rule application (step 9), we propagate back the derived concepts to the label $\mathcal{L}_{\mathbf{nfl10}}$ and detect the clash. $\diamond$

The algorithm $C_{\mathcal{T}}$ is correct: it terminates on any input and it is sound and complete.

Theorem 1 (Correctness). *Given a CKR $\mathfrak{K}$, $\mathbf{d} \in \mathfrak{D}_\Gamma$, and a concept C in NNF on the input, the tableaux algorithm $C_{\mathcal{T}}$ always terminates and C is $\mathbf{d}$-satisfiable w.r.t. $\mathfrak{K}$ iff $C_{\mathcal{T}}$ generates a complete and clash free completion tree.*

The $\mathcal{ALC}$ tableaux algorithm which we extended in this paper is in NEXPTIME [6], and this is the case also for the resulting tableaux algorithm for CKR.

Theorem 2 (Complexity). *The complexity of the $C_{\mathcal{T}}$ algorithm is* NEXPTIME *with respect to the combined size of the input.*

In general, the problem of deciding $\mathbf{d}$-satisfiability (and thus $\mathbf{d}$-subsumption) in $\mathcal{ALC}$-based CKR is EXPTIME-complete [4]. That is, the complexity is the same as for $\mathcal{ALC}$ with general TBoxes [1]. To obtain an optimal algorithm for CKR on top of $\mathcal{ALC}$ we would have to extend one of the EXPTIME algorithms for $\mathcal{ALC}$ (like, e.g., [7]). On the other hand, the algorithm presented in this paper is an important first step towards the algorithmic support for CKR based on more expressive DL (like $\mathcal{SHIQ}$ or $\mathcal{SROIQ}$), since the tableaux algorithms for these logics can be seen as extensions of the basic $\mathcal{ALC}$ algorithm on top of which we have built.

4 Algorithm Optimization

In this section we share our initial ideas about optimization of the algorithm. CKR maintains a certain level of separation between meta and object knowledge (the former influences the latter but not vice versa). Object reasoning queries the meta knowledge only to verify the coverage between dimensional vectors. The number of contexts m is typically much smaller than the size of whole KB n and the number of dimensions k

is assumed to be a constant, it hence makes sense to precompute[3] the context coverage beforehand. This can be done within $k \times m^2 = O(m^2)$ queries of the form $\mathfrak{M} \models d \prec_A d'$. Consequently, meta reasoning does not slow down object reasoning more than in other approaches with simpler meta knowledge representations [14].

One of the advantages of contextual reasoning is that the KB is split into smaller units and reasoning can be parallelized. Let us briefly sketch how this can be done with CKR. Reasoning in each context will be handled by a separate processor, which will exchange messages to deal with knowledge propagation. The computation time will be bounded by the context which requires the longest execution time together with the number of required messages. In this sense, the $\sqcup$-, $\sqcap$-, $\exists$-, $\forall$-, $\top_A$-, $\top_R$-, and $\top_{\sqsubseteq}$-rules are locally executed. The remaining rules will be implemented as follows:

$\Delta\!\uparrow$-, $\Delta\!\downarrow$-rules: the fact that a node has to be added into the target context is detected locally in the source context. A message is sent into the target context and this fact is also cached in the source context, which will be used by the other rules.

A-, R-rules: thanks to caching of the information to which contexts nodes have been added, it can be locally detected that the concept and role labels of some node have to be propagated to the target context which is then done by a message.

M-rule: note that if $a^{\mathbf{g}} \in V_{\mathbf{d}}$, $a^{\mathbf{h}} \in V_{\mathbf{e}}$, and $\mathbf{d} \prec \mathbf{e}$, then eventually $a^{\mathbf{g}}$ is added into $V_{\mathbf{e}}$ by the $\Delta\!\uparrow$-rule. Therefore also the precondition the M-rule can be locally verified and once detected, a respective message is sent to all other contexts.

Propagation of knowledge increases the number of messages and can trigger additional computation in the target context. It is hence desired to limit it to the necessary cases only. Using a technique similar to *lazy unfolding* [1], we were able to optimize the three tableaux rules $\top_A$, $\top_R$, and $\top_{\sqsubseteq}$ for propagation of the $\top_{\mathbf{e}}$ symbols as follows:

$\top^*_A$-rule: **if** $x \in V_{\mathbf{f}}$, $\mathbf{d} \preceq \mathbf{e}$, $A_{\mathbf{d}} \in \mathcal{L}_{\mathbf{f}}(x), \top_{\mathbf{e}} \notin \mathcal{L}_{\mathbf{f}}(x)$
then $\mathcal{L}_{\mathbf{f}}(x) := \mathcal{L}_{\mathbf{f}}(x) \cup \{\top_{\mathbf{e}}\}$

$\top^*_R$-rule: **if** $x, y \in V_{\mathbf{f}}$, $\mathbf{d} \preceq \mathbf{e}$, $R_{\mathbf{d}} \in \mathcal{L}_{\mathbf{f}}(\langle x, y\rangle), \top_{\mathbf{e}} \notin \mathcal{L}_{\mathbf{f}}(x) \cap \mathcal{L}_{\mathbf{f}}(y)$
then $\mathcal{L}_{\mathbf{f}}(x) := \mathcal{L}_{\mathbf{f}}(x) \cup \{\top_{\mathbf{e}}\}$, $\mathcal{L}_{\mathbf{f}}(y) := \mathcal{L}_{\mathbf{f}}(y) \cup \{\top_{\mathbf{e}}\}$

$\top^*_{\sqsubseteq}$-rule: **if** $x \in V_{\mathbf{f}}$, $\mathbf{d} \prec \mathbf{e}$, $\neg\top_{\mathbf{e}} \in \mathcal{L}_{\mathbf{f}}(x), \neg\top_{\mathbf{d}} \notin \mathcal{L}_{\mathbf{f}}(x)$
then $\mathcal{L}_{\mathbf{f}}(x) := \mathcal{L}_{\mathbf{f}}(x) \cup \{\neg\top_{\mathbf{d}}\}$

The main idea of these optimized rules is to avoid the introduction of a number of disjunctive concept expressions of the form $\neg\top_{\mathbf{d}} \sqcup \top_{\mathbf{e}}$ caused by the $\top_{\sqsubseteq}$-rule which could possibly cause unnecessary non-deterministic branching. Instead, we apply each disjunction only after one of the disjuncts is proven untrue.

Normally the $\top_A$-rule would add $\top_{\mathbf{d}}$ into the label of any node x in which $A_{\mathbf{d}}$ was found. Consequently, the $\top_{\sqsubseteq}$-rule would be fired once for each $\mathbf{e} \succ \mathbf{d}$ and add $\neg\top_{\mathbf{d}} \sqcup \top_{\mathbf{e}}$ every time. This eventually results into adding $\top_{\mathbf{e}}$ into the same label for each such $\mathbf{e}$ over the run of the algorithm. The optimized $\top^*_A$-rule skips the introduction of these disjunctions and directly adds the $\top_{\mathbf{e}}$ symbol for all such $\mathbf{e}$. The $\top_R$-rule is also optimized in the very same fashion. Hence the two optimized rules $\top^*_A$- and $\top^*_R$-rule

[3] This does not imply that a DL KB at meta level is useless. In meta knowledge modeling, DL axioms on dimensional values can constrain the coverage structure, e.g., given a location dimension, we can require cities to be located within some country: $\mathsf{City} \sqsubseteq \exists \prec_{\mathsf{location}} .\mathsf{Country}$.

do the work previously done by the $\top_A$- and $\top_R$-rules but in addition they take care of the first part of the disjunction $\neg\top_{\mathbf{d}} \sqcup \top_{\mathbf{e}}$ (i.e., the one which adds $\top_{\mathbf{e}}$ if $\top_{\mathbf{d}}$ was found). We still have to take care of the second part, and this is done by the $\top^*_{\sqsubseteq}$-rule which adds $\neg\top_{\mathbf{d}}$ to any label in which $\neg\top_{\mathbf{e}}$ was found for $\mathbf{d} \prec \mathbf{e}$.

The version of the algorithm $C_{\mathcal{T}}$ that uses the $\top^*_A$-, $\top^*_R$-, and $\top^*_{\sqsubseteq}$-rules instead of the $\top_A$-, $\top_R$-, and $\top_{\sqsubseteq}$-rules respectively, will be denoted by $C_{\mathcal{T}}{}^*$.

Theorem 3 (Correctness of optimized rules). *Given a CKR $\mathfrak{K}$, $\mathbf{d} \in \mathfrak{D}_\Gamma$, and a concept C in NNF on the input, the algorithm $C_{\mathcal{T}}{}^*$ always terminates and C is $\mathbf{d}$-satisfiable w.r.t. $\mathfrak{K}$ iff $C_{\mathcal{T}}{}^*$ generates a complete and clash free completion tree.*

Example 2 (Optimized tableaux rules). Let us now compare the original tableaux rules with the optimized rules by the following deduction:

$$\mathfrak{K}_{fb} \models \mathbf{sp} : \mathsf{TopSportsman} \sqsubseteq \forall \mathsf{playsFor}_{\mathbf{wc10}}.\mathsf{WinnerTeam}_{\mathbf{wc10}}$$

The algorithm is initialized with $V_{\mathbf{sp}} = \{s_0\}$ and the label $\mathcal{L}_{\mathbf{sp}} = \{\mathsf{TopSportsman} \sqcap \exists \mathsf{playsFor}_{\mathbf{wc10}}.\neg\mathsf{WinnerTeam}_{\mathbf{wc10}}\}$. The original algorithm $C_{\mathcal{T}}$ proceeds as follows:

(1) $\mathcal{L}_{\mathbf{sp}}(s_0) := \mathcal{L}_{\mathbf{sp}}(s_0) \cup \{\mathsf{TopSportsman}, \exists \mathsf{playsFor}_{\mathbf{wc10}}.\neg\mathsf{WinnerTeam}_{\mathbf{wc10}}\}$ by $\sqcap$-rule;
(2) $V_{\mathbf{sp}} := V_{\mathbf{sp}} \cup \{s_1\}$, $E_{\mathbf{sp}} = \{\langle s_0, s_1\rangle\}$, $\mathcal{L}_{\mathbf{sp}}(\langle s_0, s_1\rangle) = \{\mathsf{playsFor}_{\mathbf{wc10}}\}$, $\mathcal{L}_{\mathbf{sp}}(s_1) = \{\neg\mathsf{WinnerTeam}_{\mathbf{wc10}}\}$ by $\exists$-rule;
(3) $\mathcal{L}_{\mathbf{sp}}(s_0) := \mathcal{L}_{\mathbf{sp}}(s_0) \cup \{\top_{\mathbf{wc10}}\}$, $\mathcal{L}_{\mathbf{sp}}(s_1) := \mathcal{L}_{\mathbf{sp}}(s_1) \cup \{\top_{\mathbf{wc10}}\}$ by $\top_R$-rule;
(4) $\mathcal{L}_{\mathbf{sp}}(s_0) := \mathcal{L}_{\mathbf{sp}}(s_0) \cup \{\neg\top_{\mathbf{wc10}} \sqcup \top_{\mathbf{fb}}, \neg\top_{\mathbf{nfl10}} \sqcup \top_{\mathbf{fb}}, \neg\top_{\mathbf{wc10}} \sqcup \top_{\mathbf{sp}}, \neg\top_{\mathbf{nfl10}} \sqcup \top_{\mathbf{sp}}, \neg\top_{\mathbf{fb}} \sqcup \top_{\mathbf{sp}}\}$,
$\mathcal{L}_{\mathbf{sp}}(s_1) := \mathcal{L}_{\mathbf{sp}}(s_1) \cup \{\neg\top_{\mathbf{wc10}} \sqcup \top_{\mathbf{fb}}, \neg\top_{\mathbf{nfl10}} \sqcup \top_{\mathbf{fb}}, \neg\top_{\mathbf{wc10}} \sqcup \top_{\mathbf{sp}}, \neg\top_{\mathbf{nfl10}} \sqcup \top_{\mathbf{sp}}, \neg\top_{\mathbf{fb}} \sqcup \top_{\mathbf{sp}}\}$ by multiple applications of the $\top_{\sqsubseteq}$-rule;
(5) $\mathcal{L}_{\mathbf{sp}}(s_0) := \mathcal{L}_{\mathbf{sp}}(s_0) \cup \{\top_{\mathbf{fb}}, \top_{\mathbf{sp}}\}$, $\mathcal{L}_{\mathbf{sp}}(s_1) := \mathcal{L}_{\mathbf{sp}}(s_1) \cup \{\neg\top_{\mathbf{nfl10}}, \top_{\mathbf{fb}}, \top_{\mathbf{sp}}\}$ by $\sqcup$-rule;
(6) $V_{\mathbf{fb}} = \{s_0, s_1\}$, $\mathcal{L}_{\mathbf{fb}}(s_0) = \{\mathsf{TopSportsman}_{\mathbf{sp}}\}$ by $\Delta\!\downarrow$- and A-rules;
(7) $\mathcal{L}_{\mathbf{fb}}(s_0) := \mathcal{L}_{\mathbf{fb}}(s_0) \cup \{\mathsf{WorldChampionPlayer}\}$ by $\mathcal{T}$-rule[4] and $\sqcup$-rule;
(8) $\mathcal{L}_{\mathbf{fb}}(s_0) := \mathcal{L}_{\mathbf{fb}}(s_0) \cup \{\mathsf{ChampionPlayer}_{\mathbf{wc10}}\}$ by $\mathcal{T}$- and $\sqcup$-rules;
(9) $V_{\mathbf{wc10}} = \{s_0, s_1\}$, $\mathcal{L}_{\mathbf{wc10}}(s_0) = \{\mathsf{ChampionPlayer}\}$, $\mathcal{L}_{\mathbf{wc10}}(\langle s_0, s_1\rangle) = \{\mathsf{playsFor}\}$ by $\Delta\!\downarrow$- A and R-rules;
(10) $\mathcal{L}_{\mathbf{wc10}}(s_0) := \mathcal{L}_{\mathbf{wc10}}(s_0) \cup \{\forall \mathsf{playsFor}.\mathsf{WinnerTeam}\}$ by $\mathcal{T}$- and $\sqcup$-rules;
(11) $\mathcal{L}_{\mathbf{wc10}}(s_1) := \mathcal{L}_{\mathbf{wc10}}(s_1) \cup \{\mathsf{WinnerTeam}\}$ by $\forall$-rule;
(12) $\mathcal{L}_{\mathbf{sp}}(s_1) := \mathcal{L}_{\mathbf{sp}}(s_1) \cup \{\mathsf{WinnerTeam}_{\mathbf{wc10}}\}$ by A-rule;

The last rule application yields a clash since we obtain $\mathcal{L}_{\mathbf{sp}}(s_1) = \{\neg\mathsf{WinnerTeam}_{\mathbf{wc10}}, \mathsf{WinnerTeam}_{\mathbf{wc10}}\}$. Notice that out of the ten applications of the $\top_{\sqsubseteq}$-rule in step (4), only the one resulting into adding $\neg\top_{\mathbf{wc10}} \sqcup \top_{\mathbf{fb}}$ into $\mathcal{L}_{\mathbf{fb}}(s_0)$ is actually needed for propagation of the concept TopSportsman into $\mathcal{L}_{\mathbf{fb}}(s_0)$. On the other hand, the addition of $\neg\top_{\mathbf{nfl10}} \sqcup \top_{\mathbf{fb}}, \neg\top_{\mathbf{nfl10}} \sqcup \top_{\mathbf{sp}}$ into the labels of both nodes (carrying irrelevant information about the context for $\mathbf{nfl10}$) is preliminary at this point and it may lead to unnecessary choices by the $\sqcup$-rule which may need to be backtracked later on – for instance the choice to add $\neg\top_{\mathbf{nfl10}}$ to s_1 in step (5). If instead the optimized algorithm $C_{\mathcal{T}}{}^*$ is used, a similar derivation is obtained in which steps (3)–(5) are replaced with:

[4] The two disjunctive concepts $\neg\mathsf{TopSportsman}_{\mathbf{sp}} \sqcup \mathsf{WorldChampionPlayer}$ and $\neg\mathsf{WorldChampionPlayer} \sqcup \mathsf{ChampionPlayer}_{\mathbf{wc10}}$ which are added to $\mathcal{L}_{\mathbf{fb}}$ in steps (7) and (8) respectively by the $\mathcal{T}$-rule are not listed here to improve readability.

(3') $\mathcal{L}_{\mathbf{sp}}(s_0) := \mathcal{L}_{\mathbf{sp}}(s_0) \cup \{\top_{\mathbf{wc10}}, \top_{\mathbf{fb}}, \top_{\mathbf{sp}}\}$,
$\mathcal{L}_{\mathbf{sp}}(s_1) := \mathcal{L}_{\mathbf{sp}}(s_1) \cup \{\top_{\mathbf{wc10}}, \top_{\mathbf{fb}}, \top_{\mathbf{sp}}\}$ by $\top_R^*$-rule;

The remainder of derivation is the same: the unnecessary choice is thus avoided. $\diamond$

The optimized rules constrain the propagation of $\top_{\mathbf{e}}$ concepts, which are needed to reflect the context hierarchy in reasoning, to necessary propagations only and avoid the introduction of unnecessary disjunctive concepts which may cause branching. Observe that in Examples 1 and 2 we have shown the application of rules in the right order. However, additional non relevant rules may be applied by the algorithm before a clash is reached. For instance, due to the axiom $\mathsf{WinnerTeam} \equiv \mathsf{WinnerFinal}$ in $\mathrm{K}(\mathcal{C}_{\mathbf{wc10}})$ the algorithm may add $\mathsf{WinnerFinal}$ into $\mathcal{L}_{\mathbf{wc10}}(s_1)$ after step (11) in Example 2 (by $\mathcal{T}$- and $\sqcup$-rules) and consequently propagate $\mathsf{WinnerFinal}_{\mathbf{wc10}}$ into $\mathcal{L}_{\mathbf{fb}}(s_1)$ and $\mathcal{L}_{\mathbf{sp}}(s_1)$ by A-rule. Such a propagation is unnecessary as there are no axioms in $\mathcal{C}_{\mathbf{fb}}$ nor $\mathcal{C}_{\mathbf{sp}}$ which could derive new knowledge from $\mathsf{WinnerFinal}_{\mathbf{wc10}}$. Therefore in the future we would also like to investigate when it is necessary to propagate qualified symbols.

5 Related Works

The only other approach for reasoning with DL-based CKR is a translation from CKR into a single DL KB [13]. Unfortunately, this solution is not practically efficient, as the translation adds a large number of axioms in order to track complex relations between qualified symbols in a single KB. This is reflected also by a significant (cubic) blow up in the size of knowledge base after the translation. In contrast, a direct tableaux algorithm allows for more effective reasoning: local reasoning is executed in the respective part of the completion tree and only relevant consequences posed on other contexts are propagated into their respective tableaux labels, thus opening the possibility of parallelization. Our tableaux procedure is also related to the distributed tableaux algorithms for DDL [8] and P-DL [2], especially in the way how symbols are propagated between local tableaux. Apart from the fact that each of these algorithms implements a different semantics, our algorithm is also able to handle semantic dependencies between roles which is an open problem for DDL and P-DL so far.

Our newly introduced optimizations bring us near to approaches interested in parallelization of DL reasoning. One relevant approach in this area is presented in [10]. This work proposes a saturation procedure for the classification of the polynomial fragment $\mathcal{ELH}_{\mathcal{R}+}$ of OWL 2 EL, distributable among multiple processors as a concurrent algorithm. The paper also presents an implementation in the reasoner ELK together with a promising evaluation over known $\mathcal{EL}$ ontologies. Even if the scope of [10] is different from our work, it highlights some aspects that support our approach. In particular, it shows that there is interest in a parallelized vision of DL reasoning algorithms. Moreover, it suggests that the sort of knowledge distribution and independence between contexts which we point to can effectively result in promising performance improvements.

6 Conclusions

Contextualized Knowledge Repository (CKR) is a knowledge representation framework that provides a contextual layer for DL knowledge bases. The recently introduced reasoning algorithm [5, 9] for $\mathcal{ALC}$-based CKR provides the first direct tableaux decision procedure for contextualized knowledge. This solution is more effective than the previously known approaches based on reduction that lead to KB blow ups and loss of the divide-and-conquer advantage of contextual representation.

In this paper, we reviewed the algorithm and discussed on its possible optimization including dimensional structure caching, parallelization and a set of new rules that optimize the propagation of symbols among local tableaux. In the future we want to extend the algorithm towards more expressive DL such as $\mathcal{SHIQ}$ and $\mathcal{SROIQ}$ and formulate an $\mathrm{EXPTIME}$ algorithm based on the existing approaches [7]: we note that some of the optimizations (e.g. the lazy unfolding for $\top_{\sqsubseteq}$ or the precomputation of context coverage) can be easily adapted to different formulations of the algorithm. We will also study further optimizations for the propagation of qualified symbols.

Acknowledgements: This research was supported from the LiveMemories project. Martin Homola is also supported from the Slovak national project VEGA no. 1/1333/12.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The description logic handbook. Cambridge University Press (2003)
2. Bao, J., Caragea, D., Honavar, V.: A distributed tableau algorithm for package-based description logics. In: CRR 2006 (2006)
3. Bao, J., Tao, J., McGuinness, D.L.: Context representation for the semantic web. In: WebSci10 (2010)
4. Bozzato, L., Homola, M., Serafini, L.: ExpTime reasoning for contextualized $\mathcal{ALC}$. Tech. Rep. TR-FBK-DKM-2012-1, Fondazione Bruno Kessler, Trento, Italy (2012), `http://dkm.fbk.eu/index.php/Resources`
5. Bozzato, L., Homola, M., Serafini, L.: Tableaux for contextualized description logics. Submitted (2012)
6. Buchheit, M., Donini, F.M., Schaerf, A.: Decidable reasoning in terminological knowledge representation systems. J. Artif. Intell. Res. (JAIR) 1, 109–138 (1993)
7. Goré, R., Nguyen, L.: EXPTIME tableaux for $\mathcal{ALC}$ using sound global caching. In: DL2007. CEUR-WP, vol. 250, pp. 299–306. CEUR-WS.org (2007)
8. Homola, M., Serafini, L.: Augmenting subsumption propagation in distributed description logics. App. Artif. Intell. 24(1-2), 137–174 (2010)
9. Homola, M., Bozzato, L., Serafini, L.: Tableaux algorithm for reasoning with contextualized knowledge. Tech. Rep. TR-FBK-DKM-2011-1, Fondazione Bruno Kessler, Trento, Italy (2011), `http://dkm.fbk.eu/index.php/Resources`
10. Kazakov, Y., Krötzsch, M., Simancik, F.: Concurrent classification of $\mathcal{EL}$ ontologies. In: ISWC 2011. LNCS, vol. 7031, pp. 305–320. Springer (2011)
11. Klarman, S., Gutiérrez-Basulto, V.: Two-dimensional description logics for context-based semantic interoperability. In: AAAI-11. AAAI Press (2011)
12. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. Art. Int. 48(1), 1–26 (1991)
13. Serafini, L., Homola, M.: Contextualized knowledge repositories for the semantic web. J. of Web Sem., Special Issue: Reasoning with context in the Semantic Web 12 (2012)
14. Straccia, U., Lopes, N., Lukacsy, G., Polleres, A.: A general framework for representing and reasoning with annotated Semantic Web data. In: AAAI-10. AAAI Press (2010)