Contextualizing DL axioms: Formalization, a New Approach, and its Properties
Antoine Zimmermann, José Giménez-García

To cite this version:
Contextualizing DL axioms: Formalization, a New Approach, and its Properties

Antoine Zimmermann\textsuperscript{1} and José M. Giménez-García\textsuperscript{2}

\textsuperscript{1} Univ Lyon, MINES Saint-Étienne, CNRS, Laboratoire Hubert Curien
UMR 5516, F-42023 Saint-Étienne, France
antoine.zimmermann@emse.fr

\textsuperscript{2} Université de Lyon, CNRS, UMR 5516,
Laboratoire Hubert-Curien, Saint-Étienne, France
jose.gimenez.garcia@univ-st-etienne.fr

Abstract. We address the problem of providing contextual information about a logical formula (e.g., provenance, date of validity, or confidence) and representing it within a logical system. In this case, it is needed to rely on a higher order or non standard formalism, or some kind of reification mechanism. We explore the case of reification and formalize the concept of contextualizing logical statements in the case of Description Logics. Then, we define several properties of contextualization that are desirable. No previous approaches satisfy all of them. Consequently, we define a new way of contextually annotating statements. It is inspired by NdFluents, which is itself an extension of the 4dFluents approach for annotating statements with temporal context. In NdFluents, instances that are involved in a contextual statement are sliced into contextual parts, such that only parts in the same context hold relations to one another, with the goal of better preserving inferences. We generalize this idea by defining contextual parts of relations and classes. This formal construction better satisfies the properties, although not entirely. We show that it is a particular case of a general mechanism that NdFluents also instantiates, and present other variations.

Keywords: Annotations, Contexts, Metadata, Ontologies, Description Logic, Semantic Web, Reasoning, Reification

1 Introduction

The problem of being able to reason not only with logical formulas, but also about said formulas, is an old one in artificial intelligence. McCarthy \cite{1} proposed to extend first order logic by reifying context and formulas to introduce a binary predicate \texttt{ist}(\phi, c) satisfied if the formula \phi is true (ist) in the context c. However, a complete axiomatization and calculus for McCarthy’s contextual logic has never been formalized. Giunchiglia \cite{2} proposed the grouping of “local” formulas in contexts, and then using other kinds of formulas to characterize how knowledge from multiple contexts is compatible. This idea of locality+compatibility \cite{3} has led to several non standard formalisms for reasoning with multiple contexts \cite{4}.
Alternatively, the approach of annotated logic programming [5] considers that a contextual annotation is just a value in an algebraic structure (e.g., a number or a temporal interval). This idea was later applied to annotated RDF and RDFS [6, 7].

The representation of statement annotation has sometimes been thought of as a data model problem without consideration of the logical formalism behind. In particular, several proposals to extend the RDF data model in various ways for allowing annotations have been made: named graphs [8], RDF+ [9], RDF* [10], Yago Model [11]. However, the underlying data structures have not a clear formal semantics. Therefore, some authors advocate another approach to representing annotation of knowledge: reify the statement or its context and describe it within the formalism of the statement. This requires modifying the statement so as to integrate knowledge of the context or statement. Examples of such techniques are reification [12], N-Ary Relations [13], Singleton Property [14], and NdFluents [15]. This paper provides an abstraction of the reification techniques in the context of Description Logics (DLs) in the form of what we call contextualization functions. Additionally, we introduce a new technique for the representation of contextual annotations that satisfies better some desirable properties.

After introducing our notations for DLs in Sec. 2, we provide formal definitions that allow us to define verifiable properties of the reification techniques (Sec. 3). Our new technique, named NdTerms, is presented in Sec. 4, where we also prove to what extent it satisfies the properties of the previous section. Sec. 5 discuss some of the problems that may occur when combining knowledge having different annotations. In Sec. 6, we present how the other approaches fit in our formalization and why they do not satisfy well the properties. Finally, we discuss this and future work in Sec. 7.

2 Preliminaries

In this section, we introduce the notations and definitions we use in relation to Description Logics. Note that we use an extended version of DL where all terms can be used as concept names, role names, and individual names in the same ontology. Using the same name for different types of terms is known as “punning” in OWL 2 [16, Section 2.4.1]. Moreover, we allow more constructs than in OWL 2 DL and make no restriction on their use in order to show that our approach is not limited to a specific DL.

We assume that there is an infinite set of terms N. Every term is an individual. A role is either a term or, given roles R and S, R ∪ S, R ∩ S, ¬R, R−, R ∘ S and R+. A concept is either a term, or, given concepts C, D, role R, individuals u1, ..., uk, and natural number n, ⊥, ⊤, C ∪ D, C ∩ D, ∃R.C, ∀R.C, ≤ nR.C, ≥ nR.C, ¬C or {u1, ..., uk}. Finally, we also allow concept product C × D to define a role.

Interpretations are tuples \((\Delta^I, \mathcal{I}_u, \mathcal{I}_r, \mathcal{I}_c)\), where \(\Delta^I\) is a non-empty set (the domain of interpretation) and \(\mathcal{I}_u, \mathcal{I}_r,\) and \(\mathcal{I}_c\) are the interpretation functions for individuals, roles and concepts respectively such that:
when we write pairs \( \langle S \rangle \) satisfies \( C \)\( \alpha \) of an ontology consequence denoted by \( \text{Sig}(I) \) is a model of an ontology \( O \)uals, and normal font for unknown individuals and variables (wrt an ABox individuals with a contextual annotation. In the following, we slightly abuse notations by defining interpretations as (\( \Delta^F \times \Delta^E \)) = \( \Delta^E \) and interpretation of roles is inductively defined by (\( R \cup S \))\( \Delta^r \) = \( R^r \cup S^r \), (\( R \cap S \))\( \Delta^r \) = \( R^r \cap S^r \), (\( \neg R \))\( \Delta^r \) = (\( \Delta^F \times \Delta^E \)) \( \Diamond^E \), (\( R^+ \))\( \Delta^r \) = \{ (x, y) \mid \exists z (x, z) \in \Delta^F \cap (z, y) \in \Delta^E \} \} and (\( R^+ \))\( \Delta^r \) is the reflexive-transitive closure of \( R^r \) (with \( R \) and \( S \) being arbitrary roles).

for all \( A \in \mathbb{N} \), \( A^E \subseteq \Delta^E \) and interpretation of concepts is defined by \( \perp^E \) = \( \emptyset \), \( \top^E = \Delta^E \), (\( C \cup D \))\( \Delta^E \) = \( C^E \cup D^E \), (\( C \cap D \))\( \Delta^E \) = \( C^E \cap D^E \), (\( \exists R.C \))\( \Delta^E \) = \{ x \exists y. y \in C^E \wedge (x, y) \in \Delta^F \}, (\forall R.C \))\( \Delta^E \) = \{ x \forall y. (x, y) \in \Delta^F \Rightarrow y \in C^E \}, (\leq n R.C \))\( \Delta^E \) = \{ x \mid \{ y \in C^E \mid (x, y) \in \Delta^F \} \leq n \}, (\geq n R.C \))\( \Delta^E \) = \{ x \mid \{ y \in C^E \mid (x, y) \in \Delta^F \} \geq n \}, (\neg C)\( \Delta^E = \Delta^E \setminus C^E \}, \{ u_1, \ldots, u_k \} = \{ u_1^E, \ldots, u_k^E \} \}

where \( C \) and \( D \) are arbitrary concepts, \( R \) an arbitrary role, \( u_1, \ldots, u_k \) are individual names, and \( k \) and \( n \) two natural numbers.

Axions are either general concept inclusions \( C \subseteq D \), sub-role axioms \( R \subseteq r \), instance assertions \( C(a) \), or role assertions \( R(a, b) \), where \( C \) and \( D \) are concepts, \( R \) and \( S \) are roles, and \( a \) and \( b \) are individual names. An interpretation \( I \) satisfies axion \( C \subseteq D \); it satisfies \( R \subseteq r \); it satisfies \( C(a) \); and it satisfies \( R(a, b) \).

An ontology \( O \) is composed of a set of terms called the signature of \( O \) and denoted by \( \text{Sig}(O) \), and a set of axioms denoted by \( \text{Ax}(O) \). An interpretation \( I \) is a model of an ontology \( O \) iff for all \( \alpha \in \text{Ax}(O) \), \( I \models \alpha \). In this case, we write \( I \models O \).

In the rest of the paper, we will use \textbf{teletype font} to denote known individuals, and normal font for unknown individuals and variables (e.g., \text{City(babylon)} and \text{City(x)}).

### 3 Contextualization of Statements

A contextual annotation can be thought of as a set of ABox axioms that describe an individual representing the statement (the anchor) that is annotated. An annotated statement (or ontology) is the combination of a DL axiom (or DL ontology) with a contextual annotation.

**Definition 1 (Connected individuals).** Two terms \( a \) and \( b \) are connected individuals wrt an ABox \( A \) iff \( a \) and \( b \) are used as individual names in \( A \), and either

- \( a \) and \( b \) are the same term, or
there exists \( R_1, \ldots, R_n \) and \( z_1, \ldots, z_{n-1} \), such that:

- \( R_1(a, z_1) \) or \( R_1(z_1, a) \)
- \( R_i(z_{i-1}, z_i) \), or \( R_i(z_i, z_{i-1}) \), \( 2 \leq i \leq n-2 \)
- \( R_n(z_{n-1}, b) \), or \( R_n(b, z_{n-1}) \)

Example 1. If we consider the ABox \( A = \{ P(a, b), Q(c, b), S(d, e) \} \), the pairs of individuals \( \{a, b\}, \{b, c\}, \{a, c\} \), and \( \{d, e\} \) are connected individuals, but \( \{a, d\}, \{b, d\}, \{c, e\}, \{b, c\} \), and \( \{c, e\} \) are not.

**Definition 2 (Contextual annotation).** A contextual annotation \( C_a \) is an ABox with signature \( \{a\} \cup \Sigma \) where \( a \notin \Sigma \) is a distinguished term (called the anchor) and \( \Sigma \) is a DL signature such that \( \forall x \in \Sigma, \{a, x\} \) are connected individuals.

Example 2. The Abox \( C_A = \{ \text{validity}(a, t), \text{Interval}(t) \text{from}(t, 609BC), \to(t, 539BC), \text{prov}(a, w), \text{name}(w, \text{wikipedia}), \text{Wiki}(w) \} \) is a contextual annotation, where \( a \) is the anchor and \( \Sigma = \{ t, \text{Interval}, w, \text{wikipedia}, \text{wiki}, 609BC, 539BC \} \).

**Definition 3 (Annotated statement).** An annotated statement is a pair \( \langle \alpha, C_a \rangle \) such that \( \alpha \) is a description logic axiom and \( C_a \) is a contextual annotation.

Example 3. The pair \( \langle \alpha, C_A \rangle \), where \( \alpha = \text{capital(babylon, babylonianEmpire)} \) and \( C_A \) is the contextual annotation from Ex. 2, is an annotated statement.

**Definition 4 (Annotated ontology).** An annotated ontology is a pair \( \langle O, C_a \rangle \) such that \( O \) is a description logic ontology and \( C_a \) is a contextual annotation.

Each reification technique has an implicit construction plan in order to map an annotated statement to a resulting ontology. A contextualization (Def. 5) represents the procedure that generates a single DL ontology from a given annotated statement or ontology. The procedure must not lose information, especially not the annotation.

**Definition 5 (Contextualization).** A contextualization is a function \( f \) that maps each annotated statement \( \alpha_{C_a} = \langle \alpha, C_a \rangle \) to a description logic ontology \( f(\alpha_{C_a}) = \text{St}(\alpha_{C_a}) \cup \text{Cx}(\alpha_{C_a}) \) such that:

- there exists an individual \( u \) in the signature of \( \text{St}(\alpha_{C_a}) \) and of \( \text{Cx}(\alpha_{C_a}) \) such that:
  - for all \( R(a, x) \in C_a \), \( R(u, x) \in \text{Cx}(\alpha_{C_a}) \);
  - for all \( R(x, a) \in C_a \), \( R(x, u) \in \text{Cx}(\alpha_{C_a}) \);
  - for all \( C(a) \in C_a \), \( C(u) \in \text{Cx}(\alpha_{C_a}) \);
  - for all other \( \alpha \in C_a \), \( \alpha \in \text{Cx}(\alpha_{C_a}) \).
- there is an injective mapping between the signature of \( \alpha \) and the signature of \( \text{St}(\alpha_C) \).

We extend \( f \) to all annotated ontologies \( O_{C_a} = \langle O, C_a \rangle \) by defining \( f(O_{C_a}) = \bigcup_{\alpha \in O} f(\langle \alpha, C_a \rangle) \).
**Example 4.** An example contextualization function $f_{ex}$ introduces a fresh term $t$ for each annotated statement with a role assertion $R(a,b)$, where $R$, $a$, and $b$ are three terms, creates new axioms $\text{subject}(t,s)$, $\text{predicate}(t,R)$, $\text{object}(t,o)$, and finally removes the axiom $R(a,b)$. Notice that this construction requires the punning of term $R$. This function is analogous to RDF Reification. The result of this contextualization, along with Other possible known approaches, is described in Sec. 6.

Those are the only structures that we will consider in this paper. The remaining definitions are desirable properties that a contextualization should satisfy, especially if one wants it to preserve as much of the original knowledge as possible.

**Definition 6 (Soundness).** A contextualization function $f$ is sound wrt a set of annotated ontologies $\Omega$ iff for each $O_C a = \langle O, C a \rangle \in \Omega$ such that $O$ and $C a$ are consistent, then $f(O_C a)$ is consistent.

That is, a contextualization is sound if, when contextualizing a consistent ontology, the result is also consistent. This property avoids that the contextualization introduces unnecessary contradictions that would result in everything being entailed by it. Note that this requirement is not necessary in the opposite direction, i.e., if $f(O_C a)$ is consistent, it is not required that $O$ and $C$ are consistent.

**Example 5.** The contextualization function $f_{ex}$ from Ex. 4 is sound wrt the set of ontologies $\Omega$, where $\Omega \cup \{\text{subject}, \text{predicate}, \text{predicate}\} = \emptyset$.

**Definition 7 (Inconsistency preservation).** Let $f$ be a contextualization function. We say that $f$ preserves inconsistencies iff for all annotated ontologies $O_C a = \langle O, C a \rangle$, if $O$ is inconsistent then $f(O_C a)$ is inconsistent.

Inconsistency preservation means that a self-contradictory ontology in a given context is contextualized into an inconsistent ontology, such that bringing additional knowledge from other contexts would result in no more consistency. If something is inconsistent within a context, then it is not really worth to consider reasoning with this annotated ontology.

**Example 6.** The contextualization function $f_{ex}$ from Ex. 4 does not preserve inconsistencies. For instance, $\text{capitalOf}$ can be defined as irreflexive using the following axiom: $\exists \text{capitalOf}. \top \sqsubseteq \forall \text{capitalOf} \neg \bot$. Then, the axiom $\text{capitalOf}(\text{babylon}, \text{babylon})$ would make the ontology inconsistent. But when applying $f_{ex}$ the result is consistent.

**Definition 8 (Entailment preservation).** Let $f$ be a contextualization function. Given two description logic ontologies $O_1$ and $O_2$ such that $O_1 \models O_2$, we say that $f$ preserves the entailment between $O_1$ and $O_2$ iff for all contextual annotations $C a$, $f(O_1, C a) \models f(O_2, C a)$. Given a set $\mathcal{R} \subseteq \mathbb{N}^a$ of contextual annotations, if $f$ preserves all entailments between ontologies in $\mathcal{R}$, then we say that $f$ is entailment preserving for $\mathcal{R}$.
In short, a contextualization is entailment preserving if all the knowledge that could be inferred from the original ontology can also be inferred, in the same context, in the contextualized ontology.

Example 7. The contextualization function \( f_{ex} \) from Ex. 4 preserves entailments for the TBox of ontologies (because no modifications are made on its axioms), but it does not preserve entailments on role assertions. For instance, the axioms \( \text{capitalOf} \subseteq \text{cityOf} \), \( \text{capitalOf(babylon, babylonianEmpire)} \) entails \( \text{cityOf(babylon, babylonianEmpire)} \), but this inference is not preserved after applying \( f_{ex} \).

4 The \textit{NdTerms} Approach

This section defines the \textit{NdTerms} approach that extends the NdFluent proposal [15]. To this end, we assume that terms are divided into three infinite disjoint sets \( N^nc \), \( N^c \), and \( N^a \) called the \textit{non contextual terms}, the \textit{contextual terms}, and the \textit{anchor terms} respectively. We also assume that there is an injective function \( \mathfrak{A} : C \rightarrow N^a \) and for all \( C a \in C \) there is an injective function \( \text{ren}_{C a} : N^nc \rightarrow N^c \) and two terms \( \text{isContextualPartOf}, \text{isInContext} \in N^nc \). For any \( C a \), we extend \( \text{ren}_{C a} \) to axioms by defining \( \text{ren}_{C a}(\alpha) \) as the axiom built from \( \alpha \) by replacing all terms \( t \in \text{Sig}(\alpha) \) with \( \text{ren}_{C a}(t) \).

4.1 Contextualization Function in \textit{NdTerms}

The contextualization needs to combine the ontologies from the statements and the contextual annotation. However, if we naïvely make the union of the axioms, they could contradict, and it would not be possible to ensure the desired properties. For example, an ontology may restrict the size of the domain of interpretation to be of a fixed cardinality, while the contextual annotation may rely on more elements outside the local universe of this context. For this reason we use the concept or relativization: The ontology is modified in such a way that the interpretation of everything explicitly described in it is confined to a set, while external terms or constructs may have elements outside said set. Relativization has been applied in various logical settings over the past four decades (e.g., [17]) and applied to DLs and OWL [18], among others.

**Definition 9 (Relativization).** Let \( C a \) be a contextual annotation. Given an ontology \( O \), the relativization of \( O \) in \( C a \) is an ontology \( \text{Rel}_{C a}(O) \) built from \( O \) as follows:

1. \( \text{Sig}(\text{Rel}_{C a}(O)) = \text{Sig}(O) \cup T_{C a} \) where \( T_{C a} \) is a term not appearing in \( \text{Sig}(O) \);
2. for all appearances of \( T \) in an axiom of \( O \), replace \( T \) with \( T_{C a} \);
3. for all concepts \( \neg C \) appearing in an axiom of \( O \), replace it with \( \neg C \cap T_{C a} \);
4. for all roles \( \neg R \) appearing in an axiom of \( O \), replace it with \( \neg R \cap (T_{C a} \times T_{C a}) \);
5. for all concepts \( \forall R.C \) appearing in an axiom of \( O \), replace it with \( \forall R.C \cap T_{C a} \);
6. for all roles \( R^+ \) appearing in an axiom of \( O \), replace it with \( R^+ \cap T_{C a} \times T_{C a} \).
7. Additionally, for all terms $t \in \text{Sig}(O)$, the following axioms are in $\text{Rel}_{c_a}(O)$:

- $t \subseteq T_{c_a}$,
- $\top_{c_a} \subseteq T_{c_a}$,
- $\exists \cdot T \subseteq T_{c_a}$,
- $T \subseteq \forall \cdot T_{c_a}$.

The relativization of an ontology can be done systematically by relativizing its concepts and roles, which in turn can be achieved by using Def. 10.

**Definition 10 (Relativization of concepts and roles).** Given a contextual annotation $c_a$, we define a function $\text{rel}_{c_a}$ that maps concepts and roles to concepts and roles according to the rules of Items 2-6. That is, recursively:

- $\text{rel}_{c_a}(t) = t$;
- $\text{rel}_{c_a}(\{u_1, \ldots, u_k\}) = \{\text{rel}_{c_a}(u_1), \ldots, \text{rel}_{c_a}(u_k)\}$;
- $\text{rel}_{c_a}(C \cup D) = \text{rel}_{c_a}(C) \cup \text{rel}_{c_a}(D)$;
- $\text{rel}_{c_a}(C \cap D) = \text{rel}_{c_a}(C) \cap \text{rel}_{c_a}(D)$;
- $\text{rel}_{c_a}(\neg C) = \neg \text{rel}_{c_a}(C) \cap T_{c_a}$;
- $\text{rel}_{c_a}(C \times D) = \text{rel}_{c_a}(C) \times \text{rel}_{c_a}(D)$;
- $\text{rel}_{c_a}(R \sqcup S) = \text{rel}_{c_a}(R) \sqcup \text{rel}_{c_a}(S)$;
- $\text{rel}_{c_a}(R \sqcap S) = \text{rel}_{c_a}(R) \cap \text{rel}_{c_a}(S)$;
- $\text{rel}_{c_a}(R \circ S) = \text{rel}_{c_a}(R) \circ \text{rel}_{c_a}(S)$;
- $\text{rel}_{c_a}(\neg R) = \neg \text{rel}_{c_a}(R) \cap T_{c_a} \times T_{c_a}$;
- $\text{rel}_{c_a}(R^{-}) = \text{rel}_{c_a}(R)^{-}$;
- $\text{rel}_{c_a}(R^{+}) = \text{rel}_{c_a}(R)^{+} \cap T_{c_a} \times T_{c_a}$;
- $\text{rel}_{c_a}(\exists R.C) = \exists \text{rel}_{c_a}(R).\text{rel}_{c_a}(C)$;
- $\text{rel}_{c_a}(\forall R.C) = \forall \text{rel}_{c_a}(R).\text{rel}_{c_a}(C) \cap T_{c_a}$;
- $\text{rel}_{c_a}(\geq n R.C) = \geq n \text{rel}_{c_a}(R).\text{rel}_{c_a}(C)$;
- $\text{rel}_{c_a}(\leq n R.C) = \leq n \text{rel}_{c_a}(R).\text{rel}_{c_a}(C)$.

where $t$ is a term, $C, D$ are concepts, $R, S$ are roles, $u_1, \ldots, u_k$ are individuals, and $k, n$ are natural numbers.

**Example 8.** The axiom $\exists\text{capitalof}.\top \subseteq \forall\text{capitalof}^{-}.\bot$ from Ex. 6 is relativized into $\exists\text{capitalof}.\top_{c_a} \subseteq \forall\text{capitalof}^{-}.\bot \cap T_{c_a}$.

Then, the contextualization in $\text{NdTerms}$ is done by: (1) creating the replacement of the anchor using the function $\text{â}$, (2) renaming all the terms in the statement using the $\text{ren}$ function, (3) linking them to the original terms by the $\text{isContextualPartOf}$ relation, and (4) linking the renamed terms to the context using the $\text{isInContext}$ relation.

**Definition 11 (Contextualization in $\text{NdTerms}$).** Let $c_a \in c$ be any contextual annotation. Let $\alpha_{c_a} = \langle \alpha, c_a \rangle$ be an annotated statement such that the signatures of $\alpha$ and $c_a$ are in $\text{N}^\text{nc}$. We define the contextualization function $\text{fnd}_{\alpha}$ such that $\text{fnd}_{\alpha}(\alpha_{c_a}) = \text{St}(\alpha_{c_a}) \cup \text{Cx}(c_a)$ and:

- $\text{St}_{\alpha}(\alpha) = \{\text{ren}_{\alpha}(\text{Rel}_{c_a}(\alpha))\} \cup \{\text{isContextualPartOf}(\text{ren}_{\alpha}(t), t) \mid t \in \text{Sig}(\alpha)\} \cup \{\text{isInContext}(\text{ren}_{\alpha}(t), \text{â}(c_a)) \mid t \in \text{Sig}(\alpha)\}$.\n
– \text{Cx}(C \alpha) \text{ contains exactly the following axioms:}

  • for all \( R(a, x) \in C \alpha \), \( R(\exists(C \alpha), x) \in \text{Cx}(\alpha) \);
  • for all \( R(x, a) \in C \alpha \), \( R(x, \exists(C \alpha)) \in \text{Cx}(C \alpha) \);
  • for all \( C(a) \in C \alpha \), \( C(\exists(C \alpha)) \in \text{Cx}(C \alpha) \);
  • for all other axioms \( \beta \in C \alpha \), \( \beta \in \text{Cx}(C \alpha) \).

Similarly to Ex. 4, this construction requires punning, since all terms in the statement are used as individual names in the role assertion \( \text{isContextualPartOf}(\text{ren}_{C \alpha}(t), t) \).

\textbf{Example 9.} The NdTerms contextualization of our running example within the context \( C A \) of Ex. 2 contains the following axioms, where \( \text{term} @C \alpha \) is the result of the renaming function \( \text{ren}_{C \alpha}(\text{term}) \):

\begin{align*}
\text{capitalOf} @C A (@C A (\text{babylon} @C A, \text{babylonianEmpire} @C A)) \\
\text{isContextualPartOf}(\text{babylon} @C A, \text{babylon}) \\
\text{isContextualPartOf}(\text{babylonianEmpire} @C A, \text{babylonianEmpire}) \\
\text{isInContext}(\text{babylon}, \text{exampleContext}) \\
\text{isInContext}(\text{babylonianEmpire}, \text{exampleContext}) \\
\text{validity}(@\text{exampleContext}, t) \text{ etc.}
\end{align*}

\textbf{4.2 Properties of NdTerms}

This subsection presents the properties satisfied by \textit{NdTerms}. Due to space constraints, we omit the proofs of the theorems, but they are provided in the accompanying technical report [19].

The contextualization of \textit{NdTerms} is sound, but only wrt annotated ontologies that satisfy certain conditions. In order to present the conditions, we need to introduce the following definition, that is also used in several proofs of this paper.

\textbf{Definition 12 (Domain extensibility).} Let \( O \) be an ontology. A model \( I = \langle \Delta^I, \cdot^I \rangle \) of \( O \) is \textit{domain extensible} for \( O \) iff for all sets \( \Delta^+, I' = \langle \Delta^I \cup \Delta^+, \cdot^I \rangle \) is also a model of \( O \). An ontology is said to be \textit{model extensible} iff it has a model that is domain extensible.

Note that, even if the domain of interpretation of an ontology is infinite, that does not necessarily mean that its models are domain extensible. This notion is closely related to the notion of \textit{expansion} in [18] since if \( I \) is domain extensible, then one can build infinitely many expansions of it.

\textbf{Theorem 1 (Soundness of NdTerms).} Let \( C \alpha \) be a contextual annotation. If \( C \alpha \) has its signature in \( N_{nc} \) and is model extensible, then the contextualization function \( f_{C \alpha} \) is sound wrt annotated ontologies \( O_{C \alpha} = \langle O, C \alpha \rangle \), \( \text{Sig}(O) \subseteq N_{nc} \), and \( \text{Sig}(O) \cap \text{Sig}(C \alpha) = \emptyset \).
A model of the union of the of two ontologies can be made from the union of the models of the original ontologies if both ontologies are domain extensible. However, this is not a strong restriction, because the relativization of any consistent ontology is model extensible. Since NdTerms relativizes the ontology $O$ and requires the contextual annotation $C_a$ to be model extensible, it follows that a model can be made. Hence NdTerms is sound.

**Theorem 2 (Inconsistency preservation of NdTerms).** The contextualization function $f_{nd}$ preserves inconsistencies.

If the contextualization of an ontology $f_{nd}(O_{C_a})$ is consistent, then the original ontology $O$ is necessarily consistent. Because the renaming function $\text{ren}_{C_a}$ is injective, there exists an inverse function $\text{ren}_{C_a}^{-1}: \text{ren}_{C_a}(\mathbb{N}^c) \rightarrow \mathbb{N}^c$ which is itself injective, and the renaming of terms gives us a model. The relativization function $\text{rel}_{C_a}$, on the other hand, does not change the interpretation (since $\top_{I''}^{C_a} = \top_{I''}$, adding $\sqcap \top$ to a concept does not change its interpretation, and replacing $\top$ with $\top_{C_a}$ has no effect on the interpretation of the concepts or roles). Therefore, if a model satisfies the contextualization function $f_{nd}(O_{C_a})$, it also satisfies $O$, and it is not possible to have a consistent NdTerms contextualization of a non-consistent ontology.

In [15], we were only able to study entailment preservation in the limited setting of $pD^*$ entailment. Here we present a much stronger theorem for NdTerms.

**Theorem 3 (Entailment preservation of NdTerms).** Let $C_a$ be a contextual annotation, let $\Omega$ be a set of ontologies having their signatures in $\mathbb{N}^c$ and disjoint from the signature of $C_a$. If $C_a$ is model extensible, then NdTerms is entailment preserving for $\{(O, C_a)\}_{O \in \Omega}$.

If the ontology $O$ is inconsistent, the entailment is trivially preserved. If not, considering that the relativization of a consistent ontology preserves entailments, and that the function $\text{ren}_{C_a}$ is just a renaming (and “truth is invariant under change of notation” [20]), the entailments of the ontology $O$ are preserved. Moreover, all axioms in $\text{Cx}(C_a)$ are satisfied by the interpretation $I$, and whenever $\text{isContextualPartOf}(\text{ren}_{C_a}(t))$ or $\text{isInContext}(\text{ren}_{C_a}(t), \mathfrak{A}(K_a))$ are in the entailed contextualized ontology, then they are in original as well as well. Therefore, all entailments are preserved, and NdTerms satisfies this property.

### 5 Annotations in Multiple Contexts

So far, we assumed that all axioms of an ontology are annotated with the same contextual information. In this setting, the core of the contextualization function in NdTerms amounts to relativizing the axioms and renaming the terms. The renaming part may seem surprising because, as we said a couple of times already, “truth is invariant under change of notation”. However, the usefulness of the renaming part becomes apparent when we want to combine several annotated ontologies having different contextual annotations (say $C_{a_1}$ and $C_{a_2}$). In this
case, if the renaming functions \( \text{ren}_{C_{a1}} \) and \( \text{ren}_{C_{a2}} \) are mapping non contextual terms into disjoint sets of contextual terms, then the contextualization function \( f_{\text{nd}} \) ensures that any inference made in a context will not interact with the knowledge from another context. This avoids the contextualized knowledge to be inconsistent when combining statements in different contexts that contradict each others.

The properties presented in Sec. 3 require a little adaptation when applied to the multi-contextual setting. Indeed, in spite of the soundness theorem of Sec. 4.2, in the general case if a set of annotated ontologies \( \{O_i, \text{Ca}_i\}_{i \in I} \) are satisfying the constraints of Theorem 1, it is still possible that \( \bigcup_{i \in I} f_{\text{nd}}(\langle O_i, \text{Ca}_i \rangle) \) is inconsistent. We expect that the preservation of consistency can be guaranteed if all the signatures of \( \{O_i\} \) are disjoint from all the signatures of \( \{\text{Ca}_i\} \). Studying in more details the case of multiple contextual annotations is planned for future work.

6 Other Approaches

Here we briefly present the most relevant reification approaches in the Semantic Web. For all of them, the contextualization only annotates the role assertions, leaving other axioms unmodified.

As seen in Ex. 4, RDF reification replaces \( \alpha = R(x, y) \) with three new role assertions \( \text{subject}(a^{\text{Ca}}_\alpha, x) \), \( \text{predicate}(a^{\text{Ca}}_\alpha, R) \), and \( \text{object}(a^{\text{Ca}}_\alpha, y) \) and the axioms in the contextual annotation are anchored on \( a^{\text{Ca}}_\alpha \), which depends on the role assertion \( R \) and the contextual annotation \( \text{Ca} \). As shown in Ex. 6 and 7, this contextualization method preserves neither inconsistencies (in the sense of Def. 7) nor entailments on role assertions.

**Example 10.** An RDF reification contextualization of our running example within the context \( \text{Ca} \) of Ex. 2 contains the following axioms:

- subject(stbcobe,babylon)
- predicate(stbcobe, capital)
- object(stbcobe, babylonianEmpire)
- validity(stbcobe, t)
- etc.

N-Ary relations replaces \( R(x, y) \) by two role assertions \( p_1(R)(x, a^{\text{Ca}}_\alpha) \) and \( p_2(R)(a^{\text{Ca}}_\alpha, y) \), where \( R \) is a simple role assertion, and \( p_1 \) and \( p_2 \) are two injective functions with disjoint functions that map non-contextual roles to contextual roles. Alternatively, a new concept \( C_R \) is added for the role \( R \), and the following assertions are added: \( C_R(a^{\text{Ca}}_\alpha) \), \( p_1(R)(a^{\text{Ca}}_\alpha, x) \), and \( p_2(R)(a^{\text{Ca}}_\alpha, y) \).

**Example 11.** An N-Ary relations contextualization of our running example within the context \( \text{Ca} \) of Ex. 2 contains the following axioms:

- capitalOf#1(babylon, rbcobe)
- capitalOf#2(rbcobe, babylonianEmpire)
- validity(rbcobe, t)
- etc.
Singleton property is using a non-standard semantics of RDF but the same idea can be simulated with DL axioms. For each simple role axiom $R(x, y)$, the following axioms are added: $a_{\alpha}^x(x, y)$ (that is, the term for the anchor is used as a role), $\{x\} \equiv \exists a_{\alpha}^x \cdot \{y\}$ (which guarantees that the anchor property is a singleton), and $\text{singletonPropertyOf}(a_{\alpha}^x, R)$.

Example 12. A Singleton Property contextualization of our running example within the context $CA$ of Ex. 2 contains the following axioms:
- capital#1(babylon, babylonianEmpire)
- $\text{singletonPropertyOf}(\text{capital}#1, \text{capital})$
- $\text{validity}(\text{capital}#1, t)$
- etc.

The remaining approach, NdFluents, uses a similar approach as NdTerms except that it only renames the terms used as individuals and does not relativize the ontology. This ensures interesting properties wrt entailment preservation [15], but TBox axioms in different contexts are not distinguishable.

7 Discussion and Future Work

NdTerms and NdFluents are a concrete instantiations of a general approach of contextualizing (parts of) the terms in the ontology. Other instantiations would be possible, such as contextualizing role names (in a similar fashion as the singleton property), class names, or a combination of them. Then, NdTerms would be the approach where each and every term is contextualized, while in NdFluents only individuals are.

In the future, we would like to deepen the analysis of contextualization, filling gaps still present in this preliminary work. Especially, the combination of multiple annotations, or annotations of contextualized ontologies, present some interesting challenges. A more systematic comparison of the various approaches remains to be presented.

Acknowledgement: This work was partly funded by project WDAqua (H2020-MSCA-ITN-2014 #64279) and project OpenSensingCity (ANR-14-CE24-0029).

References