
Defeasibility in contextual reasoning with CKR^{*}

Loris Bozzato¹, Thomas Eiter², and Luciano Serafini¹

¹ Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy

² Institut für Informationssysteme, Technische Universität Wien,
Favoritenstraße 9-11, A-1040 Vienna, Austria

{bozzato,serafini}@fbk.eu, eiter@kr.tuwien.ac.at

Abstract. Recently, representation of context dependent knowledge in the Semantic Web has been recognized as a relevant issue and a number of logic based solutions have been proposed in this regard: among them, in our previous works we presented the Contextualized Knowledge Repository (CKR) framework.

A CKR knowledge base has a two layered structure, modelled by a global context and a set of local contexts: the global context not only contains the metaknowledge defining the properties of local contexts, but also holds the global (context independent) object knowledge that is shared by all of the local contexts. In many practical cases, however, it is desirable to leave the possibility to “override” the global object knowledge at the local level, by recognizing the axioms that can allow exceptional instances in the local contexts. This clearly requires to add a notion of non monotonicity across the global and the local parts of a CKR.

In this paper we present an extension to the semantics of CKR to introduce such notion of defeasible axioms. By extending a previously proposed datalog translation, we obtain a representation for CKR as a datalog program with negation under answer set semantics. This representation can be exploited as the basis for implementation of query answering for the proposed extension of CKR.

1 Introduction

Representation of context dependent knowledge in the Semantic Web has been recently recognized as a relevant issue that lead to a number of logic based proposals, e.g. [7–9, 13–15, 12]; in particular, the Contextualized Knowledge Repository (CKR) framework [12, 3, 2], with its latest formulation in [4], has been developed at the FBK in Trento.

A CKR knowledge base has a two layered structure: basically, it consists of a global context and a set of local contexts. The global context contains metaknowledge defining the properties of local contexts, as well as the global (context independent) object knowledge that is shared by all of the local contexts. Local contexts, on the other hand, contain knowledge that holds under specific circumstances (e.g. an event) and thus represent different independent views of the domain. The global object knowledge is propagated from the global to the local contexts and used as a common part of the system knowledge. In many practical cases, however, it is desirable to leave the possibility to “override” the global object knowledge at the local level, by recognizing those axioms that can allow exceptional instances in their local instantiations.

^{*} This paper has been previously presented at the 5th International Workshop on Acquisition, Representation and Reasoning with Contextualized Knowledge (ARCOE-LogIC 2013).

For example, in the scenario of an event recommendation system, we might want to assert at the global level that “*by default, all of the cheap events are interesting*”, but then override this inclusion for particular kind of events in the context of a participant. We might also want to express defeasibility on the propagation of information: for instance, in a CKR representing an organization, we might want to express that “*by default, all the employees working the previous year also work in the current year*” and override the axiom in the context of a specific year for employees that finished their working contract at that time. In other words, we want to specify that certain axioms at the global level are *defeasible*, thus they can allow exceptional instances in local contexts, while holding in the general case: this clearly requires to add a notion of non-monotonicity across the global and the local parts of a CKR.

In this work, we present an extension to the CKR semantics of [4] to support such defeasibility for global object knowledge. We desire to enrich previous work and to have a datalog representation of the extended CKR semantics that extends the one for the CKR semantics in [4]: we introduced defeasible axioms guided by the approach of inheritance logic programs in [5]. There the idea is that special rules recognize exceptional facts at the local level and others propagate global facts only if they are not proved to be overridden at the local level, which happens if the opposite is derived; in the same vein, we consider instances of axioms that might be overridden at the local level. The semantics for CKR we define is (as desired) representable by a datalog program with negation under answer sets semantics; furthermore, a respective translation can be used as the basis to implement query answering over defeasible CKR knowledge bases.

We can thus summarize the contributions of our work as follows. After a brief introduction of preliminary definitions in Section 2, we present in Section 3 syntax and semantics of an extension of CKR with defeasible axioms in the global context. Notably, this allow us to introduce for the first time a notion of non-monotonicity across contexts in our contextual framework. We then extend in Section 4 the datalog translation for OWL RL based CKR from [4] with rules for the translation of defeasible axioms and for considering local exceptions in the propagation of such knowledge. We express non-monotonicity using answer set semantics, such that instance checking over an CKR with defeasible axioms reduces to cautious inference from all answer set of the translation. The work reported here is in progress, and an implementation of a prototype reasoner, based on the results of [4] and this paper, is underway.

2 Preliminaries: \mathcal{SROIQ} -RL

This work basically builds on the materialization calculus for CKR on OWL RL recently proposed in [4]. The extension of the calculus that we present here is again formulated over the language \mathcal{SROIQ} -RL, which represents the restriction of \mathcal{SROIQ} to the OWL RL constructs [11]. The language is obtained by restricting the form of General Concept Inclusion axioms (GCIs) and concept equivalence of \mathcal{SROIQ} to $C \sqsubseteq D$ where C and D are concept expressions, called *left-side concept* and *right-side concept* respectively, and defined by the following grammar:

$$\begin{aligned} C &:= A \mid \{a\} \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \exists R.C_1 \mid \exists R.\{a\} \mid \exists R.\top \\ D &:= A \mid \neg C_1 \mid D_1 \sqcap D_2 \mid \exists R.\{a\} \mid \forall R.D_1 \mid \leq nR.C_1 \mid \leq nR.\top \end{aligned}$$

where A is a concept name, R is role name and $n \in \{0, 1\}$. A *both-side concept* E, F is a concept expression which is both a left- and right-side concept. TBox axioms can only take the form $C \sqsubseteq D$ or $E \equiv F$. The RBox for $SR\mathcal{OIQ}$ -RL can contain every role axiom of $SR\mathcal{OIQ}$ except for $\text{Ref}(R)$. ABox concept assertions can be only stated in the form $D(a)$, where D is a right-side concept.

3 CKR with defeasible axioms

We now introduce CKRs and extend them with primitives to express defeasible axioms. We first present the syntax and then define a model-based semantics for the interpretation of defeasible inheritance from the upper contexts.

A *Contextualized Knowledge Repository (CKR)* is a two layered structure. The upper layer consists of a knowledge base \mathfrak{G} , which describes two types of knowledge: (i) the structure and the properties of contexts of the CKR (called *meta-knowledge*), and (ii) the knowledge that is context independent, i.e., that holds in every context (called *global knowledge*). The lower layer is constituted by a set of (local) contexts; each contains (locally valid) facts and can also refer to what holds in other contexts.

Meta-Language. The meta-knowledge of a CKR is expressed by a DL language defined on a meta-vocabulary, containing the elements that define the contextual structure.

Definition 1 (Meta-vocabulary). A meta-vocabulary is a DL vocabulary Γ composed of a set NC_Γ of atomic concepts, a set NR_Γ of atomic roles, and a set NI_Γ of individual constants that are mutually disjoint and contain the following sets of symbols

1. $\mathbf{N} \subseteq \text{NI}_\Gamma$ of context names.
2. $\mathbf{M} \subseteq \text{NI}_\Gamma$ of module names.
3. $\mathbf{C} \subseteq \text{NC}_\Gamma$ of context classes, including the class Ctx .
4. $\mathbf{R} \subseteq \text{NR}_\Gamma$ of contextual relations.
5. $\mathbf{A} \subseteq \text{NR}_\Gamma$ of contextual attributes.
6. For every attribute $A \in \mathbf{A}$, a set $\mathbf{D}_A \subseteq \text{NI}_\Gamma$ of attribute values of A .

We use the role $\text{mod} \in \text{NR}_\Gamma$ defined on $\mathbf{N} \times \mathbf{M}$ to express associations between contexts and modules. The *meta-language* \mathcal{L}_Γ of a CKR is a DL language over Γ with the following syntactic conditions on the application of role restrictions: for every $\bullet \in \{\forall, \exists, \leq n, \geq n\}$, (i) for a concept $\bullet A.B$, then B is in the form $B = \{a\}$ with $a \in \mathbf{D}_A$; (ii) for a concept $\bullet \text{mod}.B$, then B is in the form $B = \{m\}$ with $m \in \mathbf{M}$.

Object Language. The context (in)dependent knowledge of a CKR is expressed via a DL language called *object-language* \mathcal{L}_Σ over an object-vocabulary $\Sigma = \text{NC}_\Sigma \uplus \text{NR}_\Sigma \uplus \text{NI}_\Sigma$. Expression in \mathcal{L}_Σ will be evaluated locally to each context, i.e., each context can interpret each symbol independently. However, sometimes one wants to constrain the meaning of a symbol in a context with the meaning of a symbol in some other context. For such external references, we extend the object language \mathcal{L}_Σ to \mathcal{L}_Σ^e with *eval expressions* of the form $\text{eval}(X, C)$, where X is a concept or role expression of \mathcal{L}_Σ and C is a concept expression of \mathcal{L}_Γ (with $C \sqsubseteq \text{Ctx}$).

Defeasible Axioms. Compared to [4], we extend the types of axioms that can appear in the global object knowledge \mathfrak{G} with defeasible axioms. Given an axiom $\alpha \in \mathcal{L}_\Sigma$, the assertion $D(\alpha)$ in \mathfrak{G} states that α is a *defeasible axiom* of \mathfrak{G} . Intuitively, this statement means that α , at the level of instantiations for individuals, propagates to a local context unless it is contradicted there, and thus an exception to α for individuals is tolerated.

Example 1. A defeasible fact $D(\text{Expensive}(\text{concert}))$ might express that a concert is expensive and propagate this to local contexts. At such a context, this might be contradicted by an assertion $\neg\text{Expensive}(\text{concert})$, which then *overrides* the global assertion. Likewise, such overriding should take place if we have a global axiom $\text{Cheap} \sqsubseteq \neg\text{Expensive}$ and a local assertion $\text{Cheap}(\text{concert})$, such that $\neg\text{Expensive}(\text{concert})$ can be derived at the local context. \diamond

Example 2. Beyond facts, from a defeasible GCI axiom $D(\text{Cheap} \sqsubseteq \text{Interesting})$ (“cheap events are interesting”) and the global assertion $\text{Cheap}(\text{fbmatch})$ (“football matches are cheap”), we may conclude $\text{Interesting}(\text{fbmatch})$ at a local context; however, an assertion $\neg\text{Interesting}(\text{fbmatch})$ there would contradict this and should override the instance of the defeasible axiom for *fbmatch*: that is, the fact $\neg\text{Cheap} \sqcup \text{Interesting}(\text{fbmatch})$ may be violated. \diamond

The DL language \mathcal{L}_{Σ}^D extends \mathcal{L}_{Σ} with the set of defeasible axioms in \mathcal{L}_{Σ} .

CKR Syntax. We are now ready to give our formal definition of Contextualized Knowledge Repository.

Definition 2 (Contextualized Knowledge Repository, CKR). A Contextualized Knowledge Repository (CKR) over a meta-vocabulary Γ and an object vocabulary Σ is a structure $\mathfrak{R} = \langle \mathfrak{G}, \{K_m\}_{m \in \mathbf{M}} \rangle$ where:

- \mathfrak{G} is a DL knowledge base over $\mathcal{L}_{\Gamma} \cup \mathcal{L}_{\Sigma}^D$, and
- every K_m is a DL knowledge base over \mathcal{L}_{Σ}^e , for each module name $m \in \mathbf{M}$.

In particular, \mathfrak{R} is a *SRIQ-RL CKR*, if \mathfrak{G} and all K_m are knowledge bases over the extended language of *SRIQ-RL* where eval-expressions can only occur in left-concepts and contain left-concepts or roles. In the following, we tacitly focus on such CKR. We show how the examples in the introduction can be formalized as CKRs.

Example 3. In the first example (inspired by a real application of CKR³) we want to define an event recommendation system: we thus represent touristic events and preferences of tourists in order to be able to derive appropriate suggestions. In particular, we want to assert that, in general, all of the *Cheap* events are *Interesting*; we do so by expressing this as a defeasible axiom in the global context. Furthermore, we propose local markets (*market*) and football matches (*fbmatch*) as examples of cheap events. On the other hand, we want to reflect that tourists interested in cultural events are not interested in a sportive event like a football match⁴; we express this by negating the interest in *fbmatch*. Thus, our example CKR $\mathfrak{R}_{\text{tour}} = \langle \mathfrak{G}, \{K_{\text{ctourist}_m}\} \rangle$ has:

$$\begin{aligned} \mathfrak{G} : & \{ D(\text{Cheap} \sqsubseteq \text{Interesting}), \text{Cheap}(\text{fbmatch}), \text{Cheap}(\text{market}), \\ & \text{mod}(\text{cultural_tourist}, \text{ctourist_m}) \}, \\ K_{\text{ctourist}_m} : & \{ \neg\text{Interesting}(\text{fbmatch}) \}. \end{aligned}$$

Note that the negative assertion in the local context represents, as discussed in Example 2, an exception to the defeasible axiom: we want to recognize this “overriding” for the *fbmatch* instance, but still apply the defeasible inclusion for *market*. \diamond

³ <http://www.investintrentino.it/News/Trentour-Trentino-platform-for-smart-tourism>

⁴ To keep things simple, we omit modeling sportive events by a separate concept.

Example 4. Our next example shows how we can represent a form of defeasible propagation of information across local contexts using *eval* expressions. We want to represent the information about an organization in a CKR, using contexts to represent its situation in different years. We express the rule that every employee working the years before (*WorkingBefore*) also works in the current year (*WorkingNow*) by a defeasible inclusion. In the module associated to 2012, we say that *alice*, *bob* and *charlie* were working last year. In the module for 2013, we say (using an *eval* expression) that all of the employees working in 2012 have to be considered in the set of employees working in the past years; moreover, we say that *charlie* no longer works for the organization. This can be encoded in the CKR $\mathfrak{R}_{org} = \langle \mathfrak{G}, \{K_{em2012.m}, K_{em2013.m}\} \rangle$, where

$$\begin{aligned} \mathfrak{G} &: \left\{ \begin{array}{l} D(\textit{WorkingBefore} \sqsubseteq \textit{WorkingNow}), \\ \text{mod}(\textit{employees2012}, \textit{em2012.m}), \text{mod}(\textit{employees2013}, \textit{em2013.m}) \end{array} \right\} \\ K_{em2012.m} &: \{ \textit{WorkingNow}(\textit{alice}), \textit{WorkingNow}(\textit{bob}), \textit{WorkingNow}(\textit{charlie}) \} \\ K_{em2013.m} &: \left\{ \begin{array}{l} \textit{eval}(\textit{WorkingNow}, \{\textit{employees2012}\}) \sqsubseteq \textit{WorkingBefore}, \\ \neg \textit{WorkingNow}(\textit{charlie}) \end{array} \right\} \end{aligned}$$

Intuitively, at the local context *employees2013*, where *WorkingBefore(charlie)* can be derived, the negative assertion $\neg \textit{WorkingNow}(\textit{charlie})$ should override the instance of the inclusion axiom in the global context for *charlie*, as it would lead to the opposite, i.e., *WorkingNow(charlie)*; on the other hand, for *alice* and *bob* no overriding should happen and we can derive that they still work for the organization. \diamond

Semantics. We now define a model-based semantics for CKRs. The idea is to model exceptions of axiom instances by so called clashing assumptions, which are pairs $\langle \alpha, \mathbf{e} \rangle$ of an axiom and a set of individuals, to the effect that in the evaluation at the local context, the instance of α for \mathbf{e} is disregarded. However, such a clashing assumption must be justified, in the sense that the instance of α for \mathbf{e} must be unsatisfiable at the local context. This is ensured if there are assertions that can be derived which prove this unsatisfiability: we call such assertions clashing sets. Models of a CKR will be then CKR interpretations in [4] extended with clashing assumptions that are all justified.

Definition 3 (CKR interpretation). A CKR interpretation for $\langle \Gamma, \Sigma \rangle$ is a structure $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t.

- \mathcal{M} is a DL interpretation of $\Gamma \cup \Sigma$ s.t., for every $c \in \mathbf{N}$, $c^{\mathcal{M}} \in \text{Ctx}^{\mathcal{M}}$ and, for every $C \in \mathbf{C}$, $C^{\mathcal{M}} \subseteq \text{Ctx}^{\mathcal{M}}$;
- for every $x \in \text{Ctx}^{\mathcal{M}}$, $\mathcal{I}(x)$ is a DL interpretation over Σ s.t. $\Delta^{\mathcal{I}(x)} = \Delta^{\mathcal{M}}$ and, for $a \in \text{NI}_{\Sigma}$, $a^{\mathcal{I}(x)} = a^{\mathcal{M}}$.

The interpretation of ordinary DL expressions on \mathcal{M} and $\mathcal{I}(x)$ in $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ is as usual; *eval* expressions are interpreted as follows: for every $x \in \text{Ctx}^{\mathcal{M}}$,

$$\textit{eval}(X, C)^{\mathcal{I}(x)} = \bigcup_{\mathbf{e} \in C^{\mathcal{M}}} X^{\mathcal{I}(\mathbf{e})}$$

According to the previous definition, a CKR interpretation consists of an interpretation for the “upper-layer” (which includes the global knowledge and the meta-knowledge) and an interpretation of the object language for each instance of type context (i.e., for all $x \in \text{Ctx}^{\mathcal{M}}$), providing a semantics of the object-vocabulary in x .

An *instantiation* of an axiom $\alpha \in \mathcal{L}_\Sigma$ with a tuple \mathbf{e} of individuals in NI_Σ , written $\alpha(\mathbf{e})$, is the specialization of α , viewed as its first order translation in an universal sentence $\forall \mathbf{x}. \phi_\alpha(\mathbf{x})$, to \mathbf{e} (i.e., $\phi_\alpha(\mathbf{e})$); accordingly, e.g., \mathbf{e} is void for assertions, a single element e for GCIs, and a pair e_1, e_2 of elements for role axioms.

A *clashing assumption* is pair $\langle \alpha, \mathbf{e} \rangle$ such that $\alpha(\mathbf{e})$ is an axiom instantiation; intuitively, it represents the assumption that $\alpha(\mathbf{e})$ is not (DL-)satisfiable. A *clashing set* for a $\langle \alpha, \mathbf{e} \rangle$ is a satisfiable set S of ABox assertions such that $S \cup \{\alpha(\mathbf{e})\}$ is unsatisfiable. That is, S provides an assertional “justification” for the assumption of local overriding of α on \mathbf{e} .

Given a CKR interpretation $\mathcal{I} = \langle \mathcal{M}, \mathcal{I} \rangle$, we call the structure $\mathcal{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, CAS \rangle$ a *CAS-interpretation*, where CAS is a map such that, for every $x \in \Delta^\mathcal{M}$, $CAS(x)$ is a set of clashing assumptions for x .

Definition 4 (CAS-model). Given a CKR $\mathfrak{R} = \langle \mathfrak{G}, \{K_m\}_{m \in \mathbf{M}} \rangle$ and a CAS-interpretation $\mathcal{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, CAS \rangle$, we say that \mathcal{I}_{CAS} is a *CAS-model* for \mathfrak{R} ($\mathcal{I}_{CAS} \models \mathfrak{R}$) if

- for every $\alpha \in \mathcal{L}_\Sigma \cup \mathcal{L}_\Gamma$ in \mathfrak{G} , $\mathcal{M} \models \alpha$;
- for every $D(\alpha) \in \mathfrak{G}$ with $\alpha \in \mathcal{L}_\Sigma$, $\mathcal{M} \models \alpha$;
- if $\langle x, y \rangle \in \text{mod}^\mathcal{M}$ and $y = m^\mathcal{M}$, then $\mathcal{I}(x) \models K_m$;
- for every $\alpha \in \mathfrak{G} \cap \mathcal{L}_\Sigma$ and $x \in \text{Ctx}^\mathcal{M}$, $\mathcal{I}(x) \models \alpha$, and
- for every $D(\alpha) \in \mathfrak{G}$ with $\alpha \in \mathcal{L}_\Sigma$, $x \in \text{Ctx}^\mathcal{M}$, and domain elements $\mathbf{d} \subseteq \Delta^{\mathcal{I}(x)}$, if $\mathbf{d} \neq \mathbf{e}^\mathcal{M}$ for every $\langle \alpha, \mathbf{e} \rangle \in CAS(x)$, then $\mathcal{I}(x) \models \alpha(\mathbf{d})$.

For $\alpha \in \mathcal{L}_\Sigma^e$ and $\mathbf{c} \in \mathbf{N}$, we write $\mathfrak{R} \models_{CAS} \mathbf{c} : \alpha$ if for every CAS-interpretation \mathcal{I}_{CAS} it holds that $\mathcal{I}_{CAS} \models \mathfrak{R}$ implies $\mathcal{I}(\mathbf{c}^\mathcal{M}) \models \alpha$; for $\alpha \in \mathcal{L}_\Gamma$, we write $\mathfrak{R} \models_{CAS} \alpha$ if for every CAS-interpretation \mathcal{I}_{CAS} it holds that $\mathcal{I}_{CAS} \models \mathfrak{R}$ implies $\mathcal{M} \models \alpha$.

We say that a CAS-model $\mathcal{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, CAS \rangle$ of \mathfrak{R} is *justified* if, for every $x \in \text{Ctx}^\mathcal{M}$ and $\langle \alpha, \mathbf{e} \rangle \in CAS(x)$, some clashing set $S_{x, \langle \alpha, \mathbf{e} \rangle}$ exists such that for every CAS-model $\mathcal{I}'_{CAS} = \langle \mathcal{M}', \mathcal{I}', CAS' \rangle$ of \mathfrak{R} with $\Delta^\mathcal{M} = \Delta^{\mathcal{M}'}$, it holds $\mathcal{I}'(x) \models S_{x, \langle \alpha, \mathbf{e} \rangle}$. Informally, justification requires that we have factual evidence that an instantiation of an axiom can not be satisfied, and this evidence is provable. Based on this, we now give the definition of CKR model:

Definition 5 (CKR model). A CKR interpretation $\mathcal{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ is a *CKR model* of \mathfrak{R} (in symbols, $\mathcal{I} \models \mathfrak{R}$), if some $\mathcal{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, CAS \rangle$ is a justified CAS-model for \mathfrak{R} .

For $\alpha \in \mathcal{L}_\Sigma^e$ and $\mathbf{c} \in \mathbf{N}$, we write $\mathfrak{R} \models \mathbf{c} : \alpha$ if $\mathcal{I}(\mathbf{c}^\mathcal{M}) \models \alpha$ for every CKR model \mathcal{I} of \mathfrak{R} ; similarly for $\alpha \in \mathcal{L}_\Gamma$, we write $\mathfrak{R} \models \alpha$ if $\mathcal{M} \models \alpha$ for every CKR model \mathcal{I} of \mathfrak{R} .

We can show the following properties:

Proposition 1. Suppose that $\mathcal{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ and $\mathcal{I}' = \langle \mathcal{M}', \mathcal{I}' \rangle$ are CKR models of \mathfrak{R} such that $\mathcal{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, CAS \rangle$ and $\mathcal{I}'_{CAS'} = \langle \mathcal{M}', \mathcal{I}', CAS' \rangle$ are justified CAS-models of \mathfrak{R} . Then $\text{Ctx}^\mathcal{M} = \text{Ctx}^{\mathcal{M}'}$ and $CAS'(x) \subseteq CAS(x)$ for every $x \in \text{Ctx}^\mathcal{M}$ implies $CAS = CAS'$.

The previous result shows that, given the justification of the CKR model, there is a notion of minimality on the sets of clashing assumptions related to each context.

Given a CAS-interpretation $\mathcal{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, CAS \rangle$, we denote with $\mathcal{I}_{CAS}^\mathbf{N} = \langle \mathcal{M}', \mathcal{I}', CAS' \rangle$ the interpretation in which: (i) $\Delta^{\mathcal{M}'} = \{\alpha^\mathcal{M} \mid \alpha \in \text{NI}_\Gamma \cup \text{NI}_\Sigma\}$; (ii) \mathcal{M}' , \mathcal{I}' and CAS' are the restrictions of \mathcal{M} , \mathcal{I} and CAS to $\Delta^{\mathcal{M}'}$, respectively.

Proposition 2. *Let \mathcal{J}_{CAS} be a justified CAS-model of \mathfrak{R} . Then, also the CAS-interpretation \mathcal{J}_{CAS}^N is a justified CAS-model of \mathfrak{R} .*

Basically, the result shows that justification for a CKR only depends on the “named contexts” part of the considered CAS-model: intuitively, this allow us to consider such restricted models in the correctness result of the datalog translation presented in the following section.

Example 5. We can now show an example of CKR models satisfying the example CKRs presented in Example 3.

In the case of \mathfrak{R}_{tour} , we can consider a model $\mathcal{J}_{CAS_{tour}} = \langle \mathcal{M}, \mathcal{I}, CAS_{tour} \rangle$ such that $CAS_{tour}(\text{cultural_tourist}^M) = \{\langle Cheap \sqsubseteq Interesting, \{fbmatch\} \rangle\}$. Note that the interpretation is justified as it is easy to check that $\mathfrak{R}_{tour} \models \text{cultural_tourist} : \{Cheap(fbmatch), \neg Interesting(fbmatch)\}$, that represents a clashing set for the defeasible axiom. Moreover, for the definition of satisfiability under the assumptions in CAS_{tour} , we obtain that $\mathcal{I}(\text{cultural_tourist}^M) \models Interesting(\text{market})$.

Similarly, for \mathfrak{R}_{org} , we have that the model $\mathcal{J}_{CAS_{org}} = \langle \mathcal{M}, \mathcal{I}, CAS_{org} \rangle$ with $CAS_{org}(\text{employees2013}^M) = \{\langle WorkingBefore \sqsubseteq WorkingNow, \{charlie\} \rangle\}$ is a CKR model for the example CKR. For the interpretation of *eval* expressions, in every interpretation of \mathfrak{R}_{org} we have that $\{alice^{\mathcal{I}(x)}, bob^{\mathcal{I}(x)}, charlie^{\mathcal{I}(x)}\} \sqsubseteq WorkingBefore^{\mathcal{I}(x)}$, where $x = \text{employees2013}$. Thus the justification of the model can be easily seen as $\mathfrak{R}_{org} \models \text{employees2013} : S$ for $S = \{WorkingBefore(charlie), \neg WorkingNow(charlie)\}$, which represents a clashing set for the defeasible axiom on *charlie*. On the other hand, for the satisfiability under the assumptions in CAS_{org} , we obtain that $\mathcal{I}(\text{employees}^M) \models WorkingNow(alice)$ and $\mathcal{I}(\text{employees}^M) \models WorkingNow(bob)$. \diamond

As clashing assumptions in CAS maps are ground instances of axioms, they refer merely to named individuals. We remark that using standard names for the domain elements, one could permit clashing assumptions for all elements; the results in Propositions 1 and 2 carry over to this setting.

4 CKR translation to general programs

We revise the datalog translation for \mathcal{SROIQ} -RL CKR from [4] with rules for the detection of axiom overriding and defeasible propagation of global knowledge. To simplify the presentation of rules, we introduce a normal form for the considered axioms. We say that a CKR $\mathfrak{R} = \langle \mathfrak{G}, \{K_m\}_{m \in M} \rangle$ is in *normal form* if:

- \mathfrak{G} contains axioms in \mathcal{L}_Γ of the form of Table 1 or in the form $C \sqsubseteq \exists \text{mod.}\{m\}$, $C \sqsubseteq \exists A.\{d_A\}$ for $A, B, C \in \mathbf{C}$, $R, S, T \in \mathbf{R}$, $a, b \in \mathbf{N}$, $m \in \mathbf{M}$, $A \in \mathbf{A}$ and $d_A \in D_A$.
- \mathfrak{G} and every K_m contain axioms in \mathcal{L}_Σ of the form of Table 1 and every K_m contain axioms in \mathcal{L}_Σ^e of the form $eval(A, C) \sqsubseteq B$, $eval(R, C) \sqsubseteq T$ for $A, B, C \in \mathbf{NC}_\Sigma$, $a, b \in \mathbf{NI}_\Sigma$, $R, S, T \in \mathbf{NR}_\Sigma$ and $C \in \mathbf{C}$.
- \mathfrak{G} contains defeasible axioms $D(\alpha) \in \mathcal{L}_\Sigma^D$ with α of the form of Table 1.

It can be seen that for named interpretations, i.e., of the form \mathcal{J}_{CAS}^N , every CKR can be rewritten into an equivalent one in normal form (using new symbols).

$A(a)$	$R(a, b)$	$\neg A(b)$	$\neg R(a, b)$	$a = b$	$a \neq b$
$A \sqsubseteq B$	$\{a\} \sqsubseteq B$	$A \sqsubseteq \neg B$	$A \sqcap B \sqsubseteq C$		
$\exists R.A \sqsubseteq B$	$A \sqsubseteq \exists R.\{a\}$	$A \sqsubseteq \forall R.B$	$A \sqsubseteq \leq 1R.B$		
$R \sqsubseteq T$	$R \circ S \sqsubseteq T$	$\text{Dis}(R, S)$	$\text{Inv}(R, S)$	$\text{Irr}(R)$	

Table 1. Normal form axioms

In this version of the translation, the definition of program has now to be adapted to the new form of the rules (admitting negative literals) and to the answer set semantics of the resulting logic program.

Syntax. A *signature* is a tuple $\langle \mathbf{C}, \mathbf{P} \rangle$ of a finite set \mathbf{C} of *constants* and a finite set \mathbf{P} of *predicates*. We assume a set \mathbf{V} of *variables*; the elements of $\mathbf{C} \cup \mathbf{V}$ are *terms*. An *atom* is of the form $p(t_1, \dots, t_n)$ where $p \in \mathbf{P}$ and t_1, \dots, t_n , are terms. A *literal* l is either a positive literal p or a negative literal $\neg p$ with p an atom and \neg the symbol of strong negation. Literals of the form $p, \neg p$ are *complementary*.

A rule r is an expression of the form

$$a \leftarrow b_1, \dots, b_k, \text{not } b_{k+1}, \dots, \text{not } b_m.$$

where a, b_1, \dots, b_m are literals and *not* is the negation as failure symbol (NAF). We denote with $\text{Head}(r)$ the head a of rule r and with $\text{Body}^+(r)$ and $\text{Body}^-(r)$ the positive (b_1, \dots, b_k) and NAF (b_{k+1}, \dots, b_m) part of the body of the rule⁵. A *fact* H is a ground rule with empty body (we then omit \leftarrow). A *program* P is a finite set of rules. A *ground substitution* σ for $\langle \mathbf{C}, \mathbf{P} \rangle$ is a function $\sigma : \mathbf{V} \rightarrow \mathbf{C}$; (ground) substitutions on atoms and *ground instances* of atoms are as usual.

Semantics. Given a program P , we define the *universe* U_P of P as the set of all constants occurring in P and the *base* B_P of P as the set of all the ground *literals* constructible from the predicates in P and the constants in U_P . An *interpretation* $I \subseteq B_P$ is a consistent subset of B_P (i.e., not containing complementary literals). We say that a literal l is *true* in I iff $l \in I$ and *false* otherwise.

Given a rule $r \in \text{ground}(P)$, the body of r is true in I if: (1) every literal in $\text{Body}^+(r)$ is true w.r.t. I , and (2) every literal in $\text{Body}^-(r)$ is false w.r.t. I . A rule r is *satisfied* in I if either the head of r is true in I or the body of r is not true in I .

An interpretation I for P is a model for P ($I \models P$), if every rule in $\text{ground}(P)$ is satisfied in I ; it is a *minimal model* for P , if no proper subset $I' \subset I$ is a model for P .

Given an interpretation I for P , the (Gelfond-Lifschitz) *reduct* of P w.r.t. I , denoted by $G_I(P)$, is the set of rules obtained from $\text{ground}(P)$ by (i) removing every rule r such that $\text{Body}^-(r) \cap I \neq \emptyset$. (ii) removing the NAF part from the bodies of the remaining rules. Then I is an *answer set* of P , if I is a minimal model of the positive version $\text{pos}(G_I(P))$ of $G_I(P)$ (i.e. the positive program obtained by considering each negative literal $\neg p(t_1, \dots, t_n)$ as a positive one with predicate symbol $\neg p$).

The following property is well-known.

Lemma 1. *If M is an answer set for P , then M is a minimal model of P .*

⁵ In the rules, we might write $(\neg)p$ to denote that the rule holds both for the positive and negative literal associated to p . This will be used only as a shortcut to simplify the presentation of rules.

Using this interpretation for our programs, we say that a literal $H \in B_P$ is a *consequence* of P and we write $P \models H$ iff for every answer set M of P we have that $M \models H$.

Translation. We now are ready to present our translation. As in [4], we basically instantiate and adapt the materialization calculus in [10] to meet the structure of CKR.

The translation is composed by the following sets: the *input translations* $I_{glob}, I_{loc}, I_D, I_{rl}$, the *deduction rules* P_{loc}, P_D, P_{rl} , and *output translation* O , such that:

- every input translation I and output translation O are partial functions (defined over axioms in normal form) while deduction rules P are sets of datalog rules;
- given an axiom or signature symbol α (and $c \in \mathbf{N}$), each $I(\alpha)$ (or $I(\alpha, c)$) is either undefined or a set of datalog facts or rules;
- given an axiom α and $c \in \mathbf{N}$, $O(\alpha, c)$ is either undefined or a single datalog fact;

We extend the definition of input translations to knowledge bases (set of axioms) S with their signature Σ , with $I(S) = \bigcup_{\alpha \in S} I(\alpha) \cup \bigcup_{s \in \Sigma, I(s) \text{ defined}} I(s)$ (similarly $I(S, c) = \bigcup_{\alpha \in S} I(\alpha, c) \cup \bigcup_{s \in \Sigma, I(s) \text{ defined}} I(s, c)$).

We briefly present the form of the different sets of translation and deduction rules involved in the translation process: the tables containing the complete set of rules can be found in the Appendix.

The set of rules in $I_{rl}(S, c)$ define the rules to translate *SRIOQ-RL* axioms and signature: for example, we have the following rule for concept inclusions: $A \sqsubseteq B \mapsto \{\text{subClass}(A, B, c)\}$. The set of rules P_{rl} are the corresponding deduction rules for axioms in *SRIOQ-RL*: for example, for atomic concept inclusions we have the rule:

$$\text{instd}(x, z, c) \leftarrow \text{subClass}(y, z, c), \text{instd}(x, y, c).$$

Global input rules of $I_{glob}(\mathfrak{G})$, basically encode the interpretation of Ctx in the global context. Similarly, local input rules $I_{loc}(K_m, c)$ and local deduction rules P_{loc} provide the translation and rules for elements of the local object language, in particular for *eval* expression: e.g., for inclusion of concepts with a left *eval* expression we have the input rule $\text{eval}(A, C) \sqsubseteq B \mapsto \{\text{subEval}(A, C, B, c)\}$ and the corresponding deduction rule

$$\text{instd}(x, b, c) \leftarrow \text{subEval}(a, c_1, b, c), \text{instd}(c', c_1, \text{gm}), \text{instd}(x, a, c').$$

The input rules in I_D provide the translation of defeasible axioms in the global context: given a defeasible axiom $D(\alpha)$, $I_{rl}(\alpha, \text{gk})$ is applied and a rule defining when the axiom is locally overridden is added to the program. For example, if $D(A \sqsubseteq B) \in \mathfrak{G}$, the fact $\text{subClass}(A, B, \text{gk})$. is added to the program for the global context together with the corresponding overriding rule:

$$\text{ovr}(\text{subClass}, x, A, B, c) \leftarrow \neg \text{instd}(x, B, c), \text{instd}(x, A, c), \text{prec}(c, g).$$

The deduction rules in P_D , on the other hand, provide the definition of the defeasible propagation of axioms from the global context to the local contexts. For example, the following rule propagates an atomic concept inclusion axiom: if the inclusion axiom is in the program of the global context and can be applied to a local instance, it is applied only if the instance is not recognized as an exception:

$$\text{instd}(x, z, c) \leftarrow \text{subClass}(y, z, g), \text{instd}(x, y, c), \\ \text{prec}(c, g), \text{not ovr}(\text{subClass}, x, y, z, c).$$

Finally, the set of output rules $O(\alpha, c)$ provides the translation of ABox assertions that can be verified by applying the rules of the final program. For example, the case for atomic concept assertions in a given context c is given as $A(a) \mapsto \{\text{instd}(a, A, c)\}$.

Given a CKR $\mathfrak{K} = \langle \mathfrak{G}, \{K_m\}_{m \in M} \rangle$, the translation to its datalog program follows these steps:

1. the *global program* for \mathfrak{G} is translated to (where gm, gk are new context names):

$$PG(\mathfrak{G}) = I_{glob}(\mathfrak{G}_\Gamma) \cup I_D(\mathfrak{G}_\Gamma) \cup I_{rl}(\mathfrak{G}_\Gamma, gm) \cup I_{rl}(\mathfrak{G}_\Sigma, gk) \cup P_{rl}$$

where $\mathfrak{G}_\Gamma = \{\alpha \in \mathfrak{G} \mid \alpha \in \mathcal{L}_\Gamma\}$ and $\mathfrak{G}_\Sigma = \{\alpha \in \mathfrak{G} \mid \alpha \in \mathcal{L}_\Sigma^D\}$.

2. We define the set of contexts:

$$\mathbf{N}_\mathfrak{G} = \{c \in \mathbf{N} \mid PG(\mathfrak{G}) \models \text{instd}(c, \text{Ctx}, gm)\}$$

For every $c \in \mathbf{N}_\mathfrak{G}$, we define its associated knowledge base as:

$$K_c = \bigcup \{K_m \in \mathfrak{K} \mid PG(\mathfrak{G}) \models \text{triple}(c, \text{mod}, m, gm)\}$$

3. We define each *local program* for $c \in \mathbf{N}_\mathfrak{G}$ as:

$$PC(c) := P_{loc} \cup P_D \cup I_{loc}(K_c, c) \cup I_{rl}(K_c, c) \cup \{\text{prec}(c, gk)\}.$$

4. The *CKR program* is then defined as:

$$PK(\mathfrak{K}) = PG(\mathfrak{G}) \cup \bigcup_{c \in \mathbf{N}_\mathfrak{G}} PC(c)$$

We say that \mathfrak{G} *entails* an axiom $\alpha \in \mathcal{L}_\Gamma$ (denoted $\mathfrak{G} \models_{\overline{P}} \alpha$) if $PG(\mathfrak{G})$ and $O(\alpha, gm)$ are defined and $PG(\mathfrak{G}) \models O(\alpha, gm)$. Similarly, \mathfrak{G} entails an axiom $\alpha \in \mathcal{L}_\Sigma$ (denoted $\mathfrak{G} \models_{\overline{P}} \alpha$) if $PG(\mathfrak{G})$ and $O(\alpha, gk)$ are defined and $PG(\mathfrak{G}) \models O(\alpha, gk)$. We say that \mathfrak{K} *entails* an axiom $\alpha \in \mathcal{L}_\Sigma^e$ in a context $c \in \mathbf{N}$ (denoted $\mathfrak{K} \models_{\overline{P}} c : \alpha$), if the elements of $PK(\mathfrak{K})$ and $O(\alpha, c)$ are defined and $PK(\mathfrak{K}) \models O(\alpha, c)$.

Correctness of the translation. In the following we show the correctness of the translation with respect to the problem of instance checking in the presented semantics for CKR. Let $CAS_{\mathbf{N}}$ be a map associating every $c \in \mathbf{N}$ to a set $CAS_{\mathbf{N}}(c)$ of clashing assumptions. We define the set of the corresponding overriding assumptions:

$$OVR(CAS_{\mathbf{N}}) = \{\text{ovr}(p(\mathbf{e})) \mid \langle \alpha, \mathbf{e} \rangle \in CAS_{\mathbf{N}}(c), I_{rl}(\alpha, c) = p\}$$

Given a CKR \mathfrak{K} and its associated CKR program $PK(\mathfrak{K})$, the *OVR-reduct* of $PK(\mathfrak{K})$ (denoted by $G_{OVR}(PK(\mathfrak{K}))$) is the set of rules obtained from $ground(PK(\mathfrak{K}))$ by: (i) removing every rule r such that $B^-(r) \cap OVR(CAS_{\mathbf{N}}) \neq \emptyset$; (ii) removing the NAF part (which involves only instances of ovr) from the bodies of the remaining rules.

Consider a CKR interpretation $\mathcal{J} = \langle \mathcal{M}, \mathcal{I} \rangle$ and $CAS_{\mathbf{N}}^{\mathcal{M}}$ such that $CAS_{\mathbf{N}}^{\mathcal{M}}(c^{\mathcal{M}}) = CAS_{\mathbf{N}}(c)$, for all $c \in \mathbf{N}$. We can show that the following property holds:

Lemma 2. $G_{OVR}(PK(\mathfrak{R})) \models O(\alpha, c)$ iff $\mathfrak{R} \models_{CAS_{\mathbf{N}}^M} c : \alpha$ (if $O(\alpha, c)$ is defined).

We can prove the lemma by establishing the following propositions

Proposition 3 (cf. [4]). For every α , $ground(PG(\mathfrak{G})) \models O(\alpha, g)$ iff $\mathfrak{G} \models \alpha$.

Proposition 4. For every α (s.t. $O(\alpha, c)$ is defined),

1. (CAS-soundness) If $G_{OVR}(PK(\mathfrak{R})) \models O(\alpha, c)$, then $\mathfrak{R} \models_{CAS_{\mathbf{N}}^M} c : \alpha$.
2. (CAS-completeness) If $\mathfrak{R} \models_{CAS_{\mathbf{N}}^M} c : \alpha$, then $G_{OVR}(PK(\mathfrak{R})) \models O(\alpha, c)$.

The completeness of the translation procedure with respect to CKR semantics can be proved using the following results. Let $S|_p$ be the restriction of an answer set S to the set of facts for predicate p .

Lemma 3. For every justified CAS-model $\mathfrak{J}_{CAS_{\mathbf{N}}^M} = \langle \mathcal{M}, \mathcal{I}, CAS_{\mathbf{N}}^M \rangle$ of \mathfrak{R} , some answer set S of $PK(\mathfrak{R})$ exists such that $S|_{ovr} = OVR(CAS_{\mathbf{N}})$.

Lemma 4. For every answer set S of $PK(\mathfrak{R})$, some justified CAS-model $\mathfrak{J}_{CAS_S^M} = \langle \mathcal{M}, \mathcal{I}, CAS_S^M \rangle$ of \mathfrak{R} exists such that $CAS_S(c) = \{\langle \alpha, e \rangle \mid I_{rl}(\alpha, c) = p, ovr(p(e)) \in S\}$ for every $c \in \mathbf{N}$.

The correctness result directly follows from previous results.

Theorem 1. For a normal form CKR \mathfrak{R} , $\mathfrak{R} \models c : \alpha$ iff $PK(\mathfrak{R}) \models O(\alpha, c)$ (provided $O(\alpha, c)$ is not void).

5 Conclusion

We presented an extension to the Contextualized Knowledge Repository (CKR) framework introducing a notion of defeasibility of axioms across contexts. We then presented a datalog translation for the extended semantics, based on the materialization calculus for instance checking in [4]. In the translation, non-monotonicity is expressed using answer set semantics, such that instance checking over OWL RL based CKR reduces to cautious inference from all answer sets of the translation.

An implementation of the presented translation in a prototype is ongoing. It basically builds on the *DReW* DL datalog rewriter [16], and extends the basic translation of OWL RL ontologies to the two layered structure of CKR. Given as input an ontology representing the global context and ontologies representing the knowledge modules, the translation produces a datalog program (compliant to DLV syntax) representing the input CKR. More specifically, it encodes the rules and the two layered translation process presented in Section 4: after the translation of the global context, the set of contexts and their associations to modules are derived from the global program through interaction with the DLV solver; with the local knowledge bases defined, the programs for the local contexts are then computed. We aim at extending the current prototype to access external data sources; moreover, we want to compare the query performance with the available implementation of CKR based on SPARQL forward rules [4].

An interesting direction for future investigation is the comparison of the proposed approach to the known approaches to integrate notions of defeasibility and defaults in description logics. Notable examples of such approaches include representations of typicality in DLs concepts [6] and works employing circumscription in description logics [1]. It is interesting to see whether such notions of defeasibility can be reduced to our representation, thus leading to an implementation of such approaches.

Another natural continuation to the presented work is to allow defeasible axioms across local contexts, possibly along an explicit order relation between contexts (as the *coverage* relation [12]), or across knowledge modules, allowing overriding in specific instances of context classes associated to such modules.

Acknowledgments. The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no.257641 (PlanetData NoE).

References

1. Bonatti, P.A., Lutz, C., Wolter, F.: Description logics with circumscription. In: KR. pp. 400–410 (2006)
2. Bozzato, L., Ghidini, C., Serafini, L.: Comparing contextual and flat representations of knowledge: a concrete case about football data. In: K-CAP 2013. pp. 9–16. ACM (2013)
3. Bozzato, L., Homola, M., Serafini, L.: Towards More Effective Tableaux Reasoning for CKR. In: DL2012. CEUR-WP, vol. 824, pp. 114–124. CEUR-WS.org (2012)
4. Bozzato, L., Serafini, L.: Materialization Calculus for Contexts in the Semantic Web. In: DL2013. CEUR-WP, vol. 1014. CEUR-WS.org (2013)
5. Buccafurri, F., Faber, W., Leone, N.: Disjunctive logic programs with inheritance. In: ICLP. pp. 79–93 (1999)
6. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: A non-monotonic description logic for reasoning about typicality. *Artif. Intell.* 195, 165–202 (2013)
7. Khriyenko, O., Terziyan, V.: A framework for context sensitive metadata description. *IJSMO* 1(2), 154–164 (2006)
8. Klarman, S., Gutiérrez-Basulto, V.: $\mathcal{ALC}_{\mathcal{ALC}}$: a context description logic. In: JELIA (2010)
9. Klarman, S.: Reasoning with Contexts in Description Logics. Ph.D. thesis, Free University of Amsterdam (2013)
10. Krötzsch, M.: Efficient Inferencing for OWL EL. In: JELIA 2010. Lecture Notes in Computer Science, vol. 6341, pp. 234–246. Springer (2010)
11. Motik, B., Fokoue, A., Horrocks, I., Wu, Z., Lutz, C., Grau, B.C.: OWL 2 Web Ontology Language Profiles. W3C recommendation, W3C (Oct 2009), <http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/>
12. Serafini, L., Homola, M.: Contextualized knowledge repositories for the semantic web. *J. of Web Semantics* 12 (2012)
13. Straccia, U., Lopes, N., Lukácsy, G., Polleres, A.: A general framework for representing and reasoning with annotated semantic web data. In: AAAI 2010. AAAI Press (2010)
14. Tanca, L.: Context-Based Data Tailoring for Mobile Users. In: BTW 2007 Workshops. pp. 282–295 (2007)
15. Udrea, O., Recupero, D., Subrahmanian, V.S.: Annotated RDF. *ACM Trans. Comput. Log.* 11(2), 1–41 (2010)
16. Xiao, G., Heymans, S., Eiter, T.: DReW: a reasoner for datalog-rewritable description logics and dl-programs. In: BuRO 2010 (2010)

A Appendix: rules tables

Global input rules $I_{glob}(\emptyset)$

- (igl-subctx1) $C \in \mathbf{C} \mapsto \{\text{subClass}(C, \text{Ctx}, \text{gm})\}$
 (igl-subctx2) $c \in \mathbf{N} \mapsto \{\text{insta}(c, \text{Ctx}, \text{gm})\}$

Local input rules $I_{loc}(K_m, c)$

- (ilc-subevalat) $\text{eval}(A, C) \sqsubseteq B \mapsto \{\text{subEval}(A, C, B, c)\}$
 (ilc-subevalr) $\text{eval}(R, C) \sqsubseteq T \mapsto \{\text{subEvalR}(R, C, T, c)\}$

Local deduction rules P_{loc}

- (plc-subevalat) $\text{instd}(x, b, c) \leftarrow \text{subEval}(a, c_1, b, c), \text{instd}(c', c_1, \text{gm}), \text{instd}(x, a, c')$.
 (plc-subevalr) $\text{triple}(x, t, y, c) \leftarrow \text{subEvalR}(r, c_1, t, c), \text{instd}(c', c_1, \text{gm}), \text{triple}(x, r, y, c')$.
 (plc-eq) $\text{eq}(x, y, c) \leftarrow \text{nom}(x, c), \text{eq}(x, y, c')$.

Output translation $O(\alpha, c)$

- (o-concept) $A(a) \mapsto \{\text{instd}(a, A, c)\}$
 (o-role) $R(a, b) \mapsto \{\text{triple}(a, R, b, c)\}$
 (o-nconcept) $\neg A(a) \mapsto \{\neg \text{instd}(a, A, c)\}$
 (o-nrole) $\neg R(a, b) \mapsto \{\neg \text{triple}(a, R, b, c)\}$
-

Table 2. Global, local and output rules

For $D(\alpha) \in S$, apply $I_{rl}(\alpha)$ and the corresponding rule in the following:

- | | | |
|---------------|------------------------------------|---|
| (ovr-inst1) | $D(A(a))$ | $\mapsto \{\text{ovr}(\text{insta}, a, A, c) \leftarrow \neg \text{instd}(a, A, c), \text{prec}(c, g).\}$ |
| (ovr-inst2) | $D(\neg A(a))$ | $\mapsto \{\text{ovr}(\neg \text{insta}, a, A, c) \leftarrow \text{instd}(a, A, c), \text{prec}(c, g).\}$ |
| (ovr-triple1) | $D(R(a, b))$ | $\mapsto \{\text{ovr}(\text{triplea}, a, R, b, c) \leftarrow \neg \text{triple}(a, R, b, c), \text{prec}(c, g).\}$ |
| (ovr-triple2) | $D(\neg R(a, b))$ | $\mapsto \{\text{ovr}(\neg \text{triplea}, a, R, b, c) \leftarrow \text{triple}(a, R, b, c), \text{prec}(c, g).\}$ |
| (ovr-inst3) | $D(\{a\} \sqsubseteq B)$ | $\mapsto \{\text{ovr}(\text{insta}, a, B, c) \leftarrow \neg \text{instd}(a, B, c), \text{prec}(c, g).\}$ |
| (ovr-subc) | $D(A \sqsubseteq B)$ | $\mapsto \{\text{ovr}(\text{subClass}, x, A, B, c) \leftarrow \neg \text{instd}(x, B, c), \text{instd}(x, A, c), \text{prec}(c, g).\}$ |
| (ovr-not) | $D(A \sqsubseteq \neg B)$ | $\mapsto \{\text{ovr}(\text{supNot}, x, A, B, c) \leftarrow \text{instd}(x, B, c), \text{instd}(x, A, c), \text{prec}(c, g).\}$ |
| (ovr-cnj) | $D(A_1 \sqcap A_2 \sqsubseteq B)$ | $\mapsto \{\text{ovr}(\text{subConj}, x, A_1, A_2, B, c) \leftarrow \neg \text{instd}(x, B, c), \text{instd}(x, A_1, c), \text{instd}(x, A_2, c), \text{prec}(c, g).\}$ |
| (ovr-subex) | $D(\exists R.A \sqsubseteq B)$ | $\mapsto \{\text{ovr}(\text{subEx}, x, R, A, B, c) \leftarrow \neg \text{instd}(x, B, c), \text{triple}(x, R, y, c), \text{instd}(y, A, c), \text{prec}(c, g).\}$ |
| (ovr-supex) | $D(A \sqsubseteq \exists R.\{a\})$ | $\mapsto \{\text{ovr}(\text{supEx}, x, A, R, a, c) \leftarrow \neg \text{triple}(x, R, a, c), \text{instd}(x, A, c), \text{prec}(c, g).\}$ |
| (ovr-forall) | $D(A \sqsubseteq \forall R.B)$ | $\mapsto \{\text{ovr}(\text{supForall}, x, y, A, R, B, c) \leftarrow \neg \text{instd}(y, B, c), \text{instd}(x, A, c), \text{triple}(x, R, y, c), \text{prec}(c, g).\}$ |
| (ovr-leqone) | $D(A \sqsubseteq \leq 1R.B)$ | $\mapsto \{\text{ovr}(\text{supLeqOne}, x, y, z, A, R, B, c) \leftarrow \neg \text{eq}(z, y, c), \text{instd}(x, A, c), \text{triple}(x, R, y, c), \text{triple}(x, R, z, c), \text{instd}(y, B, c), \text{instd}(z, B, c), \text{prec}(c, g).\}$ |
| (ovr-subr) | $D(R \sqsubseteq S)$ | $\mapsto \{\text{ovr}(\text{subRole}, x, y, R, S, c) \leftarrow \neg \text{triple}(x, S, y, c), \text{triple}(x, R, y, c), \text{prec}(c, g).\}$ |
| (ovr-subrc) | $D(R \circ S \sqsubseteq T)$ | $\mapsto \{\text{ovr}(\text{subRChain}, x, y, z, R, S, T, c) \leftarrow \neg \text{triple}(x, T, z, c), \text{triple}(x, R, y, c), \text{triple}(y, S, z, c), \text{prec}(c, g).\}$ |
| (ovr-dis) | $D(\text{Dis}(R, S))$ | $\mapsto \{\text{ovr}(\text{dis}, x, y, R, S, c) \leftarrow \text{triple}(x, S, y, c), \text{triple}(x, R, y, c), \text{prec}(c, g).\}$ |
| (ovr-inv) | $D(\text{Inv}(R, S))$ | $\mapsto \{\text{ovr}(\text{inv}, x, y, R, S, c) \leftarrow \neg \text{triple}(x, S, y, c), \text{triple}(x, R, y, c), \text{prec}(c, g).\}$ |
| (ovr-irr) | $D(\text{Irr}(R))$ | $\mapsto \{\text{ovr}(\text{irr}, x, R, c) \leftarrow \text{triple}(x, R, x, c), \text{prec}(c, g).\}$ |
-

Table 3. Input rules $I_D(S)$ for defeasible axioms

(prop-inst)	$(\neg)\text{instd}(x, z, c)$	$\leftarrow (\neg)\text{insta}(x, z, g), \text{prec}(c, g), \text{not ovr}((\neg)\text{insta}, x, z, c).$
(prop-triple)	$(\neg)\text{triple}(x, r, y, c)$	$\leftarrow (\neg)\text{triplea}(x, r, y, g), \text{prec}(c, g), \text{not ovr}((\neg)\text{triplea}, x, r, y, c).$
(prop-subc)	$\text{instd}(x, z, c)$	$\leftarrow \text{subClass}(y, z, g), \text{instd}(x, y, c), \text{prec}(c, g), \text{not ovr}(\text{subClass}, x, y, z, c).$
(prop-not)	$\neg\text{instd}(x, z, c)$	$\leftarrow \text{supNot}(y, z, g), \text{instd}(x, y, c), \text{prec}(c, g), \text{not ovr}(\text{supNot}, x, y, z, c).$
(prop-cnj)	$\text{instd}(x, z, c)$	$\leftarrow \text{subConj}(y_1, y_2, z, g), \text{instd}(x, y_1, c), \text{instd}(x, y_2, c),$ $\text{prec}(c, g), \text{not ovr}(\text{subConj}, x, y_1, y_2, z, c).$
(prop-subex)	$\text{instd}(x, z, c)$	$\leftarrow \text{subEx}(v, y, z, g), \text{triple}(x, v, x', c), \text{instd}(x', y, c),$ $\text{prec}(c, g), \text{not ovr}(\text{subEx}, x, v, y, z, c).$
(prop-supex)	$\text{triple}(x, r, x', c)$	$\leftarrow \text{supEx}(y, r, x', g), \text{instd}(x, y, c),$ $\text{prec}(c, g), \text{not ovr}(\text{supEx}, x, y, r, x', c).$
(prop-forall)	$\text{instd}(y, z', c)$	$\leftarrow \text{supForall}(z, r, z', g), \text{instd}(x, z, c), \text{triple}(x, r, y, c),$ $\text{prec}(c, g), \text{not ovr}(\text{supForall}, x, y, z, r, z', c).$
(prop-leqone)	$\text{eq}(x_1, x_2, c)$	$\leftarrow \text{supLeqOne}(z, r, z', g), \text{instd}(x, z, c), \text{triple}(x, r, x_1, c),$ $\text{instd}(x_1, z', c), \text{triple}(x, r, x_2, c), \text{instd}(x_2, z', c),$ $\text{prec}(c, g), \text{not ovr}(\text{supLeqOne}, x, x_1, x_2, z, r, z', c).$
(prop-subr)	$\text{triple}(x, w, x', c)$	$\leftarrow \text{subRole}(v, w, g), \text{triple}(x, v, x', c),$ $\text{prec}(c, g), \text{not ovr}(\text{subRole}, x, y, v, w, c).$
(prop-subrc)	$\text{triple}(x, w, z, c)$	$\leftarrow \text{subRChain}(u, v, w, g), \text{triple}(x, u, y, c), \text{triple}(y, v, z, c),$ $\text{prec}(c, g), \text{not ovr}(\text{subRChain}, x, y, z, u, v, w, c).$
(prop-dis1)	$\neg\text{triple}(x, v, y, c)$	$\leftarrow \text{dis}(u, v, g), \text{triple}(x, u, y, c),$ $\text{prec}(c, g), \text{not ovr}(\text{dis}, x, y, u, v, c).$
(prop-dis2)	$\neg\text{triple}(x, u, y, c)$	$\leftarrow \text{dis}(u, v, g), \text{triple}(x, v, y, c),$ $\text{prec}(c, g), \text{not ovr}(\text{dis}, x, y, u, v, c).$
(prop-inv1)	$\text{triple}(y, v, x, c)$	$\leftarrow \text{inv}(u, v, g), \text{triple}(x, u, y, c),$ $\text{prec}(c, g), \text{not ovr}(\text{inv}, x, y, u, v, c).$
(prop-inv2)	$\text{triple}(y, u, x, c)$	$\leftarrow \text{inv}(u, v, g), \text{triple}(x, v, y, c),$ $\text{prec}(c, g), \text{not ovr}(\text{inv}, x, y, u, v, c).$
(prop-irr)	$\neg\text{triple}(x, u, x, c)$	$\leftarrow \text{irr}(u, g), \text{nom}(x, c),$ $\text{prec}(c, g), \text{not ovr}(\text{irr}, x, u, c).$

Table 4. Deduction rules P_D for defeasible axioms

RL input translation $I_{rl}(S, c)$

(irl-nom)	$a \in NI \mapsto \{\text{nom}(a, c)\}$	(irl-not)	$A \sqsubseteq \neg B \mapsto \{\text{supNot}(A, B, c)\}$
(irl-cls)	$A \in NC \mapsto \{\text{cls}(A, c)\}$	(irl-subcnj)	$A_1 \sqcap A_2 \sqsubseteq B \mapsto \{\text{subConj}(A_1, A_2, B, c)\}$
(irl-rol)	$R \in NR \mapsto \{\text{rol}(R, c)\}$	(irl-subex)	$\exists R. A \sqsubseteq B \mapsto \{\text{subEx}(R, A, B, c)\}$
(irl-inst1)	$A(a) \mapsto \{\text{insta}(a, A, c)\}$	(irl-supex)	$A \sqsubseteq \exists R. \{a\} \mapsto \{\text{supEx}(A, R, a, c)\}$
(irl-inst2)	$\neg A(a) \mapsto \{\neg \text{insta}(a, A, c)\}$	(irl-forall)	$A \sqsubseteq \forall R. B \mapsto \{\text{supForall}(A, R, B, c)\}$
(irl-triple)	$R(a, b) \mapsto \{\text{triplea}(a, R, b, c)\}$	(irl-leqone)	$A \sqsubseteq \leq 1R. B \mapsto \{\text{supLeqOne}(A, R, B, c)\}$
(irl-ntriple)	$\neg R(a, b) \mapsto \{\neg \text{triplea}(a, R, b, c)\}$	(irl-subr)	$R \sqsubseteq S \mapsto \{\text{subRole}(R, S, c)\}$
(irl-eq)	$a = b \mapsto \{\text{eq}(a, b, c)\}$	(irl-subrc)	$R \circ S \sqsubseteq T \mapsto \{\text{subRChain}(R, S, T, c)\}$
(irl-neq)	$a \neq b \mapsto \{\neg \text{eq}(a, b, c)\}$	(irl-dis)	$\text{Dis}(R, S) \mapsto \{\text{dis}(R, S, c)\}$
(irl-inst3)	$\{a\} \sqsubseteq B \mapsto \{\text{insta}(a, B, c)\}$	(irl-inv)	$\text{Inv}(R, S) \mapsto \{\text{inv}(R, S, c)\}$
(irl-subc)	$A \sqsubseteq B \mapsto \{\text{subClass}(A, B, c)\}$	(irl-irr)	$\text{Irr}(R) \mapsto \{\text{irr}(R, c)\}$
(irl-top)	$\top(a) \mapsto \{\text{insta}(a, \text{top}, c)\}$		
(irl-bot)	$\perp(a) \mapsto \{\text{insta}(a, \text{bot}, c)\}$		

RL deduction rules P_{rl}

(prl-instd)	$(\neg) \text{instd}(x, z, c)$	$\leftarrow (\neg) \text{insta}(x, z, c).$
(prl-tripled)	$(\neg) \text{tripled}(x, r, y, c)$	$\leftarrow (\neg) \text{triplea}(x, r, y, c).$
(prl-eq1)	$\text{eq}(x, x, c)$	$\leftarrow \text{nom}(x, c).$
(prl-eq2)	$(\neg) \text{eq}(y, x, c)$	$\leftarrow (\neg) \text{eq}(x, y, c).$
(prl-eq3)	$(\neg) \text{instd}(y, z, c)$	$\leftarrow \text{eq}(x, y, c), (\neg) \text{instd}(x, z, c).$
(prl-eq4)	$(\neg) \text{tripled}(y, u, z, c)$	$\leftarrow \text{eq}(x, y, c), (\neg) \text{triplea}(x, u, z, c).$
(prl-eq5)	$(\neg) \text{tripled}(z, u, y, c)$	$\leftarrow \text{eq}(x, y, c), (\neg) \text{triplea}(z, u, x, c).$
(prl-eq6)	$(\neg) \text{eq}(x, z, c)$	$\leftarrow (\neg) \text{eq}(x, y, c), (\neg) \text{eq}(y, z, c).$
(prl-top)	$\text{instd}(x, \text{top}, c)$	$\leftarrow \text{instd}(x, z, c).$
(prl-subc)	$\text{instd}(x, z, c)$	$\leftarrow \text{subClass}(y, z, c), \text{instd}(x, y, c).$
(prl-not)	$\neg \text{instd}(x, z, c)$	$\leftarrow \text{supNot}(y, z, c), \text{instd}(x, y, c).$
(prl-subcnj)	$\text{instd}(x, z, c)$	$\leftarrow \text{subConj}(y_1, y_2, z, c), \text{instd}(x, y_1, c), \text{instd}(x, y_2, c).$
(prl-subex)	$\text{instd}(x, z, c)$	$\leftarrow \text{subEx}(v, y, z, c), \text{triplea}(x, v, x', c), \text{instd}(x', y, c).$
(prl-supex)	$\text{triplea}(x, r, x', c)$	$\leftarrow \text{supEx}(y, r, x', c), \text{instd}(x, y, c).$
(prl-supforall)	$\text{instd}(y, z', c)$	$\leftarrow \text{supForall}(z, r, z', c), \text{instd}(x, z, c), \text{triplea}(x, r, y, c).$
(prl-leqone)	$\text{eq}(x_1, x_2, c)$	$\leftarrow \text{supLeqOne}(z, r, z', c), \text{instd}(x, z, c), \text{triplea}(x, r, x_1, c),$ $\text{instd}(x_1, z', c), \text{triplea}(x, r, x_2, c), \text{instd}(x_2, z', c).$
(prl-subr)	$\text{triplea}(x, w, x', c)$	$\leftarrow \text{subRole}(v, w, c), \text{triplea}(x, v, x', c).$
(prl-subrc)	$\text{triplea}(x, w, z, c)$	$\leftarrow \text{subRChain}(u, v, w, c), \text{triplea}(x, u, y, c), \text{triplea}(y, v, z, c).$
(prl-dis1)	$\neg \text{triplea}(x, v, y, c)$	$\leftarrow \text{dis}(u, v, c), \text{triplea}(x, u, y, c).$
(prl-dis2)	$\neg \text{triplea}(x, u, y, c)$	$\leftarrow \text{dis}(u, v, c), \text{triplea}(x, v, y, c).$
(prl-inv1)	$\text{triplea}(y, v, x, c)$	$\leftarrow \text{inv}(u, v, c), \text{triplea}(x, u, y, c).$
(prl-inv2)	$\text{triplea}(y, u, x, c)$	$\leftarrow \text{inv}(u, v, c), \text{triplea}(x, v, y, c).$
(prl-irr)	$\neg \text{triplea}(x, u, x, c)$	$\leftarrow \text{irr}(u, c), \text{nom}(x, c).$

Table 5. RL input and deduction rules