

All But Not Nothing: Left-Hand Side Universals for Tractable OWL Profiles

David Carral¹, Adila Krisnadhi^{1,2}, Sebastian Rudolph³, and Pascal Hitzler¹

¹ Wright State University, Dayton, OH, U.S.A.

² Universitas Indonesia, Depok, Indonesia

³ Technische Universität Dresden, Germany

Abstract. We show that occurrences of the universal quantifier in the left-hand side of general concept inclusions can be rewritten into \mathcal{EL}^{++} axioms under certain circumstances. I.e., this intuitive modeling feature is available for OWL EL while retaining tractability. Furthermore, this rewriting makes it possible to reason over corresponding extensions of \mathcal{EL}^{++} and Horn-*SR \mathcal{OIQ}* using standard reasoners.

1 Introduction

“All OWLED 2004 paper were written by George Lucas.”

The sentence just given is *vacuously* true since the OWLED conference series was established in 2005, i.e., the statement is about the empty set: There are no OWLED 2004 papers, hence all of them (namely, none) were written by George Lucas.

Vacuously true statements arise from the formal semantics under which the universal quantifier is interpreted in first-order predicate logic (and, more generally, in most of mathematics and computer science). This formal semantics gives rise to a neat mathematical theory (e.g., the universal quantifier is then just the dual of the existential quantifier), but at the same time this reading of the universal quantifier remains rather unintuitive and also an educational obstacle in the teaching of logic-based methods. In the Semantic Web realm, this was e.g. prominently pointed out in [13], but examples of course abound.

Natural language use often does not intend vacuous truth. If John says “all my children are adopted,” then we assume that he does in fact have children, all of which are adopted. If we later found out that he never had any children, then we may assess that he was lying earlier, rather than understand that his statement was vacuously true.

More formally speaking, and in terms of description logic modeling, our intuitive understanding would be that John belongs to the class

$$(\forall \text{hasChild. AdoptedPerson}) \sqcap (\exists \text{hasChild. AdoptedPerson}),$$

rather than simply $\forall \text{hasChild. AdoptedPerson}$ which would seem more similar to the natural language sentence on the syntax level. For lack of a better word, we call such an occurrence of a universal a *witnessed* universal.

The observations just given are of course not at all new. And in fact, from a formal logical perspective the observations do not even appear to be very interesting at first sight: A witnessed universal simply seems to be syntactic sugar because it can be expressed using a conjunction of a universal and an existential.

However, it turns out that the discussion becomes much more interesting in the context of logics – in this case description logics of course – where the use of universal quantifiers is restricted in some way, i.e., for Horn Description Logics and for some of the tractable OWL Profiles. Indeed, a version of witnessed universal was investigated in [11],¹ where it was shown that adding left-hand side witnessed universals to Horn-*SRIQ* does not increase the data complexity. The result was shown by a rather involved argument using a modification of the *SRIQ* tableau algorithm.

However, we noticed that it is not necessary to modify the underlying reasoning algorithm, but that instead a simple knowledge base transformation can be applied with which reasoning can be reduced to the case of knowledge bases without universals. As a result, it is possible to add left-hand side witnessed universals to OWL EL while retaining tractability. While we will later provide a formal definition and correctness proof for the transformation, let us already give an example which explains the rewriting. We want to model the statement that a Pizza all of which toppings are vegetarian, is in fact a vegetarian Pizza. Note, though, that the axiom $\forall \text{hasTopping.VegTopping} \sqsubseteq \text{VegPizza}$ does *not* sufficiently capture the intended meaning, since a Pizza without any toppings (i.e., dough only) does hardly qualify as a Pizza in most people’s minds. An adequate axiomatization thus seems to be

$$\text{Pizza} \sqcap (\forall \text{hasTopping.VegTopping}) \sqcap (\exists \text{hasTopping.VegTopping}) \sqsubseteq \text{VegPizza} \quad (1)$$

where the left-hand side does not suffer from the just mentioned vacuous truth issue.

The rewriting which we introduce in this paper now rewrites this axiom into the three axioms

$$\begin{aligned} \text{Pizza} \sqcap \exists \text{hasTopping.VegTopping} &\sqsubseteq \exists R_{\exists \text{hasTopping}}^{\text{Pizza}} \cdot \top \\ R_{\exists \text{hasTopping}}^{\text{Pizza}} &\sqsubseteq \text{hasTopping} \\ \exists R_{\exists \text{hasTopping}}^{\text{Pizza}} \cdot \text{VegTopping} &\sqsubseteq \text{VegPizza} \end{aligned}$$

where $R_{\exists \text{hasTopping}}^{\text{Pizza}}$ is a fresh role name. These axioms are expressible in \mathcal{ELH} .

So in this paper, we report on this transformation, its correctness, and some of its consequences. The plan of the paper is as follows. We first introduce some formal preliminaries in Section 2. Then, in Section 3, we show our transformation and prove it correct. We then make a proposal for an OWL/RDF and functional style syntax for witnessed universals in Section 4, before we conclude in Section 5.

¹ The conference presentation given for this paper was actually what made us start thinking about the topic.

2 Preliminaries

In the following paragraphs we formally introduce the syntax, terminology and semantics of the description logics \mathcal{EL}^{++} and Horn- \mathcal{SROIQ} as required for the paper. We assume a certain familiarity with the topic and refer the reader to the literature [1,3] if further background is required. To improve the readability of the paper and the clarity of our formal arguments we make use of succinct normal forms to describe the syntax of the aforementioned description logics. This is without loss of generality: polynomial algorithms to normalize \mathcal{EL}^{++} and Horn- \mathcal{SROIQ} ontologies can be found in [7,8].

A *DL signature* is a tuple $\langle \mathbf{N}_C, \mathbf{N}_R, \mathbf{N}_I \rangle$ where \mathbf{N}_C , \mathbf{N}_R and \mathbf{N}_I are finite (but large enough) disjoint sets of *concept names*, *role names* and *individuals* such that $\{\perp, \top\} \subseteq \mathbf{N}_C$. For the rest of this paper we assume that a signature Σ has been fixed and so omit further references to it.

Let $\mathbf{N}_R \cup \{R^- \mid R \in \mathbf{N}_R\}$ be the set of *roles*. Roles that are not elements of \mathbf{N}_R are called *inverse roles*. A *concept expression* is a conjunction $C_1 \sqcap \dots \sqcap C_n$ such that all $C_{(i)}$ are of the form $\exists R.A$, $\exists R.\text{Self}$, $\text{ran}(R)$, $\forall R.A$, $\leq 1 R.A$, A or $\{a\}$ where $A \in \mathbf{N}_C$, $R \in \mathbf{N}_R$ and $a \in \mathbf{N}_I$. Concept expressions of the form $\{a\}$ are called *nominals*.

We define a function Inv that maps every role to its inverse counterpart, as follows.

$$\text{Inv}(R) = \begin{cases} R^-, & \text{if } R \in \mathbf{N}_R \\ S, & R = S^- \end{cases}$$

2.1 Syntax

A *DL axiom* is an expression as depicted in Figure 1. Given a set \mathcal{O} of DL axioms, we define the following notions.

- Let $\sqsubseteq_{\mathcal{O}}$ be the minimal relation over roles such that $R \sqsubseteq_{\mathcal{O}} S$ if $R \sqsubseteq S$ or $\text{Inv}(R) \sqsubseteq \text{Inv}(S)$ are in \mathcal{O} .
- Let $\sqsubseteq_{\mathcal{O}}^*$ be the reflexive transitive closure of $\sqsubseteq_{\mathcal{O}}$.
- Let $\prec_{\mathcal{O}}$ be the relation over roles such that $R \prec_{\mathcal{O}} S$ and $\text{Inv}(R) \prec_{\mathcal{O}} S$ if and only if
 - $R \neq S$,
 - $R \neq \text{Inv}(S)$, and
 - $R \sqsubseteq S$, $R \circ V \sqsubseteq S$ or $V \circ R \sqsubseteq S$ are in \mathcal{O} .
- Let $\prec_{\mathcal{O}}^*$ be the transitive closure of $\prec_{\mathcal{O}}$.

A role R is *simple with respect to* \mathcal{O} if there is no axiom of the form $T \circ V \sqsubseteq S \in \mathcal{O}$ for every role S such that $S \sqsubseteq_{\mathcal{O}}^* R$ or $S \sqsubseteq_{\mathcal{O}}^* \text{Inv}(R)$.

The DL \mathcal{EL}^{++} . An \mathcal{EL}^{++} *axiom* is an expression as depicted in the first four columns of Figure 1. An \mathcal{EL}^{++} *ontology* \mathcal{O} is a set of \mathcal{EL}^{++} axioms that satisfies the following conditions.

$$\begin{array}{ccccccc}
A_1 \sqcap \dots \sqcap A_n \sqsubseteq B & \exists R.A \sqsubseteq B & A \sqsubseteq \exists S.B & R \sqsubseteq S & R \sqsubseteq S^- \\
A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B & \exists R.\text{Self} \sqsubseteq B & A \sqsubseteq \exists S.\text{Self} & R \circ V \sqsubseteq S & A \sqsubseteq \leq 1 S.B \\
\text{ran}(R) \sqsubseteq B & \{a\} \sqsubseteq B & A \sqsubseteq \{b\} & &
\end{array}$$

Fig. 1. DL Axioms. Where $A_{(i)}, B \in \mathbf{N}_C$, $R, S, V \in \mathbf{N}_R$ and $a, b \in \mathbf{N}_I$.

1. If $\exists R.\text{Self} \sqsubseteq B \in \mathcal{O}$, then R is simple.
2. If $\{R \circ V \sqsubseteq S, \text{ran}(S') \sqsubseteq B\} \subseteq \mathcal{O}$ and $S \sqsubseteq_{\mathcal{O}}^* S'$ then $\text{ran}(V) \sqsubseteq B \in \mathcal{O}$.

For a more comprehensive explanation of these restrictions, necessary to preserve tractability of reasoning algorithms, see [2].

The DL Horn-SROIQ. A *Horn-SROIQ axiom* is an expression as depicted Figure 1. A *Horn-SROIQ ontology* \mathcal{O} is a set of Horn-SROIQ axioms that satisfies the following conditions:

1. If $\exists R.\text{Self} \sqsubseteq B$ or $A \sqsubseteq \leq 1 R.B$ are in \mathcal{O} , then R is simple.
2. The relation $\prec_{\mathcal{O}}^*$ is irreflexive.

Condition (2) is commonly referred as *role regularity*. For a more comprehensive explanation of these restrictions, necessary to preserve decidability of reasoning algorithms, see [5].

Note that we allow for axioms of the form $A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B$ in our definition of \mathcal{EL}^{++} and Horn-SROIQ. These axioms are not usually considered part of the definition of these languages [2,8], and they cannot be normalized away using previous standard techniques.

2.2 Semantics

The semantics of an ontology is given through the definition of an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a set and $\cdot^{\mathcal{I}}$ is a function that maps each individual in the ontology to an element of $\Delta^{\mathcal{I}}$, each concept name to a subset of $\Delta^{\mathcal{I}}$ and each role name to a binary relation over $\Delta^{\mathcal{I}}$. Interpretations are extended to concept expressions as shown in the upper part of Figure 2. We say that an axiom α is *entailed* by an interpretation \mathcal{I} , written as $\mathcal{I} \models \alpha$, if the conditions defined in the lower part of Figure 2 hold. An interpretation \mathcal{I} *satisfies* an ontology \mathcal{O} , written as $\mathcal{I} \models \mathcal{O}$, if $\mathcal{I} \models \alpha$ for every $\alpha \in \mathcal{O}$. If this is the case, we say that \mathcal{I} is a *model* of \mathcal{O} . An ontology \mathcal{O} *entails* an axiom α , written as $\mathcal{O} \models \alpha$, if $\mathcal{I} \models \alpha$ for every model \mathcal{I} of \mathcal{O} .

3 Witnessed Left-Hand Side Universals

We first define our new transformation that rewrites axioms of the form $A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B$ into standard \mathcal{EL}^{++} (and also Horn-SROIQ) axioms.

Constructor	Syntax	Semantics
Conjunction	$C_1 \sqcap \dots \sqcap C_n$	$C_1^{\mathcal{I}} \cap \dots \cap C_n^{\mathcal{I}}$
Existential Restriction	$\exists R.A$	$\{x \mid \exists y : (x, y) \in R^{\mathcal{I}}, y \in D^{\mathcal{I}}\}$
Self Restriction	$\exists R.\text{Self}$	$\{x \mid (x, x) \in R^{\mathcal{I}}\}$
Range Restriction	$\text{ran}(R)$	$\{x \mid (y, x) \in R^{\mathcal{I}}\}$
Universal Restriction	$\forall R.A$	$\{x \mid \forall y : (x, y) \in R^{\mathcal{I}}, y \in D^{\mathcal{I}}\}$
At Most Restriction	$\leq 1 R.A$	$\{x \mid \#\{(x, y) \in R^{\mathcal{I}} \mid y \in A^{\mathcal{I}}\} \leq 1\}$
Nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
Top	\top	$\Delta^{\mathcal{I}}$
Bottom	\perp	\emptyset
Role Chain	$R \circ V$	$\{(x, z) \mid (x, y) \in R^{\mathcal{I}}, (y, z) \in V^{\mathcal{I}}\}$
Inverse Role	R	$\{(x, y) \mid (y, x) \in \text{Inv}(R)^{\mathcal{I}}\}$

Axiom	Syntax	Semantics
Concept Inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
Role Inclusion	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
Role Chain Inclusion	$R \circ V \sqsubseteq S$	$(R \circ V)^{\mathcal{I}} \subseteq S^{\mathcal{I}}$

Fig. 2. Semantics of DL Constructors and Axioms. Where $x_{(i)}, y, z \in \Delta^{\mathcal{I}}$, $C_{(i)}, D$ are concept expressions, $A \in \mathbf{N}_C$, $R_{(i)}, S \in \mathbf{R}$, $a, b \in \mathbf{N}_I$ and $v \in \mathbf{N}_V$

Definition 1. Let \mathcal{O} be an \mathcal{EL}^{++} or Horn-SROIQ ontology. Then \mathcal{O}_{\forall} is the ontology that results from substituting every axiom $\alpha = A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B \in \mathcal{O}$ by axioms $\exists R.A_2 \sqsubseteq X_{\alpha}$, $A_1 \sqcap X_{\alpha} \sqsubseteq Y_{\alpha}$, $Y_{\alpha} \sqsubseteq \exists Z_{\alpha}.\top$, $Z_{\alpha} \sqsubseteq R$ and $\exists Z_{\alpha}.A_3 \sqsubseteq B$ where X_{α} and Y_{α} are fresh concepts and Z_{α} is a fresh role.

We can verify that the transformation removes all universals while the output stays within the base language.

Lemma 1. If \mathcal{O} is an \mathcal{EL}^{++} (respectively, Horn-SROIQ) ontology, then \mathcal{O}_{\forall} is an \mathcal{EL}^{++} (respectively, Horn-SROIQ) ontology without universal quantifiers.

Proof. It is straightforward to see that the axioms added by the transformation are \mathcal{EL}^{++} (respectively, Horn-SROIQ) axioms. Also, note that the newly added axioms by the transformation do never violate any of the defining conditions for language membership. In particular, not that a role $R \in \mathbf{N}_R$ is simple with respect to \mathcal{O}_{\forall} if and only if is simple with respect to \mathcal{O} . Furthermore, note that no axiom of the form $R \circ V \sqsubseteq S$, $A \sqsubseteq \leq 1 R.B$ or $\exists R.\text{Self} \sqsubseteq B$ is added by the transformation. \square

Hence, after the transformation we can always make use of existing reasoning algorithms and tools for the appropriate description logic fragment. Also note that the number of axioms added by the transformation is linear and thus there are no implications for the complexity. Our central technical result is the following.

Theorem 1. \mathcal{O} and \mathcal{O}_\forall are equisatisfiable.

Proof. We first show how to construct a model \mathcal{J} of \mathcal{O}_\forall given a model \mathcal{I} of \mathcal{O} . Let $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$ and $\cdot^{\mathcal{J}}$ be as follows:

- For every $a \in \mathbf{N}_1$, $a^{\mathcal{I}} = a^{\mathcal{J}}$.
- For every $C \in \mathbf{N}_C$, $x \in C^{\mathcal{J}}$ if and only if $x \in C^{\mathcal{I}}$.
- For every $R \in \mathbf{N}_R$, $(x, y) \in R^{\mathcal{J}}$ if and only if $(x, y) \in R^{\mathcal{I}}$.
- For every $\alpha = A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B \in \mathcal{O}$ and every $x \in (\exists R.A_2)^{\mathcal{I}}$, $x \in X_\alpha^{\mathcal{J}}$.
- For every $\alpha = A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B \in \mathcal{O}$ and every $x \in A_1^{\mathcal{I}} \cap (\exists R.A_2)^{\mathcal{I}}$, $x \in Y_\alpha^{\mathcal{J}}$ and
 - if $x \in (\forall R.A_3)^{\mathcal{I}}$ then $(x, y) \in Z_\alpha^{\mathcal{J}}$ for some y such that $(x, y) \in R^{\mathcal{I}}$,
 - otherwise $(x, y) \in Z_\alpha^{\mathcal{J}}$ for some y such that $(x, y) \in R^{\mathcal{I}}$ and $y \notin A_3^{\mathcal{I}}$.

Note that, given a model \mathcal{I} , if some element $x \in (A_1^{\mathcal{I}} \cap (\exists R.A_2)^{\mathcal{I}}) \setminus (\forall R.A_3)^{\mathcal{I}}$ then there always exists some $y \in \Delta^{\mathcal{I}}$ such that $(x, y) \in R^{\mathcal{I}}$ and $y \notin A_3^{\mathcal{I}}$.

We proceed to verify that \mathcal{J} is indeed a model of \mathcal{O}_\forall . Note that, by definition, \mathcal{J} is identical to \mathcal{I} when considering symbols that occur in \mathcal{O} . Consequently, all axioms that are in both \mathcal{O} and \mathcal{O}_\forall are automatically entailed by \mathcal{J} as they are also entailed by model \mathcal{I} . To verify that \mathcal{J} is a model, we prove that all types of axioms in \mathcal{O}_\forall which are not in \mathcal{O} are indeed entailed by \mathcal{J} . The different types of axioms are as follows.

1. $C_1 \sqcap \dots \sqcap C_n \sqsubseteq D \in \mathcal{O}_\forall \setminus \mathcal{O}$. Then $C_1 \sqcap \dots \sqcap C_n \sqsubseteq D$ is of the form $A_1 \sqcap X_\alpha \sqsubseteq Y_\alpha$ where $\alpha = A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B$. Let $x \in A_1^{\mathcal{J}} \cap X_\alpha^{\mathcal{J}}$. Then $x \in A_1^{\mathcal{I}} \cap \exists R.A_2^{\mathcal{I}}$ by the definition of \mathcal{J} . Then $x \in Y_\alpha^{\mathcal{J}}$ also by the definition of \mathcal{J} .
2. $V \sqsubseteq S \in \mathcal{O}_\forall \setminus \mathcal{O}$. Then $V \sqsubseteq S$ is of the form $Z_\alpha \sqsubseteq R$ where $\alpha = A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B \in \mathcal{O}$. Let $(x, y) \in Z_\alpha^{\mathcal{J}}$. Then $(x, y) \in R^{\mathcal{J}}$ by the definition of \mathcal{J} .
3. $\exists S.A \sqsubseteq D \in \mathcal{O}_\forall \setminus \mathcal{O}$. Two cases arise:
 - $\exists S.A \sqsubseteq D$ is of the form $\exists R.A_2 \sqsubseteq X_\alpha$ where $\alpha = A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B \in \mathcal{O}$. Let $(x, y) \in R^{\mathcal{J}}$ and $y \in A_2^{\mathcal{J}}$. Then $x \in (\exists R.A_1)^{\mathcal{I}}$ and $x \in X_\alpha^{\mathcal{J}}$ by the definition of \mathcal{J} .
 - $\exists S.A \sqsubseteq D$ is of the form $\exists Z_\alpha.A_3 \sqsubseteq B$ where $\alpha = A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B \in \mathcal{O}$. Let $(x, y) \in Z_\alpha^{\mathcal{J}}$ and $y \in A_3^{\mathcal{J}}$. Then, by the definition of \mathcal{J} , $x \in A_1^{\mathcal{I}} \cap (\exists R.A_2)^{\mathcal{I}} \cap (\forall S.A_3)^{\mathcal{I}}$. Since \mathcal{I} is a model and $\alpha \in \mathcal{O}$ we have that $x \in B^{\mathcal{I}}$. Consequently, $x \in B^{\mathcal{J}}$ by the definition of \mathcal{J} .
4. $A \sqsubseteq \exists R.B \in \mathcal{O}_\forall \setminus \mathcal{O}$. Then $A \sqsubseteq \exists R.B$ is of the form $Y_\alpha \sqsubseteq \exists Z_\alpha.\top$. Let $x \in Y_\alpha^{\mathcal{J}}$. Then $(x, y) \in Z_\alpha$ for some y by the definition of \mathcal{J} .

We now show that every model \mathcal{J} of \mathcal{O}_\forall is a model \mathcal{I} of \mathcal{O} . Note that for every axiom $\alpha \in \mathcal{O}$ that is not of the form $A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B$ we have that $\mathcal{I} \models \alpha$ since $\alpha \in \mathcal{O}_\forall$. Hence, it suffices to show that all axioms of the form $A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B$ are indeed entailed by \mathcal{I} .

Let $\alpha = A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B \in \mathcal{O}$ and $x \in A_1^{\mathcal{I}} \cap (\exists R.A_2)^{\mathcal{I}} \cap (\forall R.A_3)^{\mathcal{I}}$. Then $x \in A_1^{\mathcal{J}} \cap (\exists R.A_2)^{\mathcal{J}} \cap (\forall R.A_3)^{\mathcal{J}}$. Note that $\{\exists R.A_2 \sqsubseteq X_\alpha, A_1 \sqcap X_\alpha \sqsubseteq Y_\alpha, Y_\alpha \sqsubseteq \exists Z_\alpha.\top, Z_\alpha \sqsubseteq R, \exists Z_\alpha.A_3 \sqsubseteq B\}$. Since \mathcal{J} is a model, we have that

$x \in X_\alpha^{\mathcal{J}} \cap Y_\alpha$ and $(x, y) \in Z_\alpha^{\mathcal{J}} \cap R^{\mathcal{J}}$ for some y . Since $x \in (\forall R.A_2)^{\mathcal{J}}$ we have that $y \in A_2^{\mathcal{J}}$. Thus, $x \in B^{\mathcal{J}}$. Since $\mathcal{I} = \mathcal{J}$, $x \in B^{\mathcal{I}}$ and $\mathcal{I} \models \exists R.A_1 \sqcap \forall R.A_2 \sqsubseteq B$. \square

Note that solving other reasoning tasks for \mathcal{EL}^{++} and Horn-*SR**OIQ* ontologies can be reduced to checking satisfiability. Therefore, our rewriting can also be used to solve other tasks such as instance retrieval or classification.

4 OWL Syntax Extension

We propose an inclusion of the witnessed universal construct into OWL 2 through several changes on the grammars of the different OWL 2 syntaxes. Before we proceed, however, note that the transformation formulated in Section 3 was defined for axioms with witnessed universal of the form $A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B$ where R, A_1, A_2, A_3 and B are all atomic. This is nevertheless not a strict limitation since arbitrary OWL 2 axioms in which such a witnessed universal occurs within a subexpression and some of R, A_1, A_2, B are not atomic can still be normalized so that witnessed universals only occur in axioms of the form either $A_1 \sqcap \exists R.A_2 \sqcap \forall R.A_3 \sqsubseteq B$ where R, A, A_1, A_2, A_3, B are all atomic. This normalization is a simple variant of the standard structural transformation that iteratively introduces definitions for non-atomic subexpressions in the original axioms [6].

More importantly, this flexibility affords us to consider extending OWL 2 syntax with witnessed universals as a type of class expression, instead of having to define a new type of OWL 2 axiom. The new syntactic element for witnessed universal we shall present in this section is first and foremost just as a syntactic shortcut for the corresponding OWL 2 class expression that uses conjunction, existential quantification, and universal quantification. Obviously, universal quantification is *not* allowed in the tractable OWL 2 EL profile. With the result from Section 3, we can now incorporate some form of universal quantification via the new syntactic element, which we shall define in the rest of this section.

4.1 OWL 2 Functional-Style Syntax Modification

We start with the modification on the grammar of OWL 2 functional-style syntax in Section 8.2 of [10]. Here, we introduce a new nonterminal symbol **ObjectSomeAllValuesFrom** to represent a witnessed universal as a new type of object property restrictions. Concretely, we first add the following production rule to the grammar:

ObjectSomeAllValuesFrom := 'ObjectSomeAllValuesFrom' '('
ObjectPropertyExpression ClassExpression ClassExpression')'

We then add the nonterminal symbol **ObjectSomeAllValuesFrom** to the **ClassExpression** production rule:

```

ClassExpression := Class | ObjectIntersectionOf | ObjectUnionOf |
    ObjectComplementOf | ObjectOneOf | ObjectSomeValuesFrom |
    ObjectAllValuesFrom | ObjectHasValue | ObjectHasSelf |
    ObjectMinCardinality | ObjectMaxCardinality |
    ObjectExactCardinality | DataSomeValuesFrom |
    DataAllValuesFrom | DataHasValue | DataMinCardinality |
    DataMaxCardinality | DataExactCardinality |
    ObjectSomeAllValuesFrom

```

The **ObjectSomeAllValuesFrom** production rule above allows one to express an object property restriction `ObjectSomeAllValuesFrom(OPE CE1 CE2)`, which is a syntactic shortcut for the class expression

```

ObjectIntersectionOf( ObjectSomeValuesFrom( OPE CE1 )
    ObjectAllValuesFrom( OPE CE2 ) ).

```

where both instances of the non-terminal OPE map to the same object property.

As an example, one can write axiom (1) from Section 1 using the OWL 2 functional-style syntax as follows:

```

SubClassOf(
    ObjectIntersectionOf( a:Pizza
        ObjectSomeAllValuesFrom( a:hasTopping a:VegTopping a:VegTopping ) )
    a:VegPizza )

```

4.2 OWL 2 Manchester Syntax Modification

For the OWL 2 Manchester syntax [4], we only need to modify the production rule for **restriction** into the rule below, using a new reserved word *someall*. As in the functional-style syntax, the expression `(P someall C D)` is a syntactic

shortcut to the expression: (P some C) and (P only D).

```

restriction ::= objectPropertyExpression 'some' primary
| objectPropertyExpression 'only' primary
| objectPropertyExpression 'value' individual
| objectPropertyExpression 'Self'
| objectPropertyExpression 'min' nonNegativeInteger [ primary ]
| objectPropertyExpression 'max' nonNegativeInteger [ primary ]
| objectPropertyExpression 'exactly' nonNegativeInteger [ primary ]
| dataPropertyExpression 'some' dataPrimary
| dataPropertyExpression 'only' dataPrimary
| dataPropertyExpression 'value' literal
| dataPropertyExpression 'min' nonNegativeInteger [ dataPrimary ]
| dataPropertyExpression 'max' nonNegativeInteger [ dataPrimary ]
| dataPropertyExpression 'exactly' nonNegativeInteger [ dataPrimary ]
| objectPropertyExpression 'someall' primary primary

```

Translation from Manchester syntax to functional-style syntax (and back) is straightforward as in Section 4.2 and 4.3 of [4]. In particular, we add the following row to the second table in Section 4.2 of the aforementioned reference.

Nonterminal	Form	Transformation (T)
restriction	<i>objectPropertyExpression</i> someall <i>primary primary</i>	ObjectSomeAllValuesFrom(T(<i>objectPropertyExpression</i>) T(<i>primary</i>) T(<i>primary</i>))

4.3 Mapping to RDF Graphs in Turtle

We add the witnessed universal construct to the mappings from functional-style syntax to RDF graphs in triple notation (Turtle), whose original document is given in [12]. For brevity, we do not directly specify the OWL/XML and RDF/XML serializations of witnessed universals, and simply assume that such serializations are straightforward to obtain from the triple notation.

To add the witnessed universal construct to the mappings, we first add the following row to the mapping from functional-style syntax to Turtle in Table 1 of [12].

Element E of the Structural Specification	Triples Generated in an Invocation T(E)	Main Node of T(E)
ObjectSomeAllValuesFrom(OPE CE ₁ CE ₂)	_:x rdf:type owl:Restriction . _:x owl:onProperty T(OPE) . _:x owl:someAllValuesFrom T(SEQ CE ₁ CE ₂) .	_:x

In the reverse mapping from Turtle to functional-style syntax, we add the following row to Table 13 in [12].

If G contains this pattern...	...then CE(...x) is set to this class expression
<pre> .:x rdf:type owl:Restriction . .:x owl:onProperty y . .:x owl:someAllValuesFrom T(SEQ z1 z2) . </pre>	<pre> ObjectSomeAllValuesFrom(OPE(y) CE(z1) CE(z2)) </pre>

4.4 OWL 2 EL Syntax Modification

According to the formal grammar of OWL 2 EL (Section 2.2 in [9]), both of the production rules for **subClassExpression** and **superClassExpression** produce the same set of class expression via the nonterminal **ClassExpression**. The result in Section 3, however, implies that witnessed universal can only appear as a subclass expression in a class inclusion axiom in order to be included to OWL 2 EL profile. This means that the production rules for **subClassExpression** and **superClassExpression** need to produce different set of class expressions, resulting in the following modified rules.

```

subClassExpression := Class | subObjectIntersectionOf | ObjectOneOf |
  subObjectSomeValuesFrom | ObjectHasValue | ObjectHasSelf |
  ObjectSomeAllValuesFrom | DataSomeValuesFrom | DataHasValue
subObjectIntersectionOf := 'ObjectIntersectionOf' '('
  subClassExpression subClassExpression { subClassExpression } ')'
subObjectSomeValuesFrom := 'ObjectSomeValuesFrom' '('
  ObjectPropertyExpression subClassExpression ')'
ObjectSomeAllValuesFrom := 'ObjectSomeAllValuesFrom' '('
  ObjectPropertyExpression subClassExpression subClassExpression ')'

superClassExpression := Class | superObjectIntersectionOf | ObjectOneOf |
  superObjectSomeValuesFrom | ObjectHasValue | ObjectHasSelf |
  DataSomeValuesFrom | DataHasValue
superObjectIntersectionOf := 'ObjectIntersectionOf' '('
  superClassExpression superClassExpression { superClassExpression } ')'
superObjectSomeValuesFrom := 'ObjectSomeValuesFrom' '('
  ObjectPropertyExpression superClassExpression ')'

```

Notice that the difference between the production rules for **subClassExpression** and **superClassExpression** only lies on the introduction of **ObjectSomeAllValuesFrom**. On the other hand, the production rules for **EquivalentClasses** and **DisjointClasses** axioms need not be changed, i.e., they still use the original production rule for **ClassExpression**.

5 Conclusions

While universal quantification is, in general, prohibited in \mathcal{EL}^{++} and Horn-*SROIQ*, we show that an intuitively rather appealing form of it, witnessed universal quantification, can be added to both of these languages without jeopardizing their pleasant computational properties. In particular, we show that their left-hand side versions can be rewritten into the base languages.

We also provided a suggestion for an OWL syntax extension which could be used as a basis for tool support for the new construct.

Acknowledgement This work was supported by the National Science Foundation under award 1017225 “III: Small: TROn – Tractable Reasoning with Ontologies” and the “La Caixa” foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, second edn. (2007)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the EL envelope. In: Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05). pp. 364–369. Morgan-Kaufmann Publishers, Edinburgh, UK (2005)
3. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
4. Horridge, M., Patel-Schneider, P. (eds.): OWL 2 Web Ontology Language Manchester Syntax (Second Edition). W3C Working Group Note (11 December 2012), available at <http://www.w3.org/TR/owl2-manchester-syntax/>
5. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible *SROIQ*. In: Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006). pp. 57–67. AAAI Press (2006)
6. Kazakov, Y.: *RIQ* and *SROIQ* are harder than *SHOIQ*. In: Brewka, G., Lang, J. (eds.) Proc. 11th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR’08). pp. 274–284. AAAI Press (2008)
7. Krötzsch, M.: Efficient inferencing for OWL EL. In: Janhunen, T., Niemelä, I. (eds.) Proc. 12th European Conf. on Logics in Artificial Intelligence (JELIA’10). LNAI, vol. 6341, pp. 234–246. Springer (2010)
8. Krötzsch, M., Rudolph, S., Hitzler, P.: Complexities of Horn description logics. ACM Trans. Comp. Log. 14(1), 2:1–2:36 (2013)
9. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C. (eds.): OWL 2 Web Ontology Language Profiles (Second Edition). W3C Recommendation (27 October 2012), available at <http://www.w3.org/TR/owl2-profiles/>
10. Motik, B., Patel-Schneider, P., Parsia, B. (eds.): OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition). W3C Recommendation (11 December 2012), available at <http://www.w3.org/TR/owl2-syntax/>

11. Nguyen, L.A., Nguyen, T.B.L., Szalas, A.: HornDL: An expressive Horn description logic with PTime data complexity. In: Faber, W., Lembo, D. (eds.) Web Reasoning and Rule Systems – 7th International Conference, RR 2013, Mannheim, Germany, July 27-29, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7994, pp. 259–264. Springer (2013)
12. Patel-Schneider, P., Motik, B. (eds.): OWL 2 Web Ontology Language Mapping to RDF Graphs (Second Edition). W3C Recommendation (11 December 2012), available at <http://www.w3.org/TR/owl2-mapping-to-rdf/>
13. Rector, A.L., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C.: OWL Pizzas: Practical experience of teaching OWL-DL: Common errors & common patterns. In: Motta, E., Shadbolt, N., Stutt, A., Gibbins, N. (eds.) Engineering Knowledge in the Age of the Semantic Web, 14th International Conference, EKAW 2004, Whittlebury Hall, UK, October 5-8, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3257, pp. 63–81. Springer (2004)