

# Description Logic Programs: Normal Forms<sup>\*</sup>

Pascal Hitzler<sup>1</sup> and Andreas Eberhart<sup>2</sup>

<sup>1</sup> AIFB, University of Karlsruhe

<sup>2</sup> Hewlett-Packard Germany

**Abstract.** The relationship and possible interplay between different knowledge representation and reasoning paradigms is a fundamental topic in artificial intelligence. For expressive knowledge representation for the Semantic Web, two different paradigms – namely Description Logics (DLs) and Logic Programming – are the two most successful approaches. A study of their exact relationships is thus paramount.

An intersection of OWL with (function-free non-disjunctive) Datalog, called DLP (for *Description Logic Programs*), has been described in [1, 2]. We provide normal forms for DLP in Description Logic syntax and in Datalog syntax, thus providing a bridge for the researcher and user who is familiar with either of these paradigms. We argue that our normal forms are the most convenient way to *define* DLP for teaching and dissemination purposes.

## 1 Introduction

The Web Ontology Language OWL<sup>3</sup> [3] has been recommended by the W3C consortium as a standard for the Semantic Web. Based on Description Logics [4], it provides a sound foundation for the development of sophisticated Semantic Web technology. It is however understood that the expressivity of OWL lacks certain features which can most naturally be covered by rule-based approaches akin to Logic Programming [5], like F-Logic and its variants [6]. At the same time, pure Logic Programming based approaches to ontology modelling are also being used in practice, in particular in the form of F-Logic. Providing interoperability between the two paradigms is thus of practical importance.

In [1] the intersection between OWL-DL and function-free disjunctive Datalog has been described, and called *Description Logic Programs* (DLP). Since then, this paradigm has been extended considerably. Most notably, it has been developed into an efficient and flexible reasoning system using techniques from disjunctive Datalog for OWL-DL reasoning [7–10] — including the translation of a major fragment of OWL-DL to disjunctive Datalog. But it has also been used in different contexts, e.g. for defining OWL Light<sup>-</sup> [11].

---

<sup>\*</sup> The authors acknowledge support by the German Federal Ministry for Education and Research under the SmartWeb project (01 IMD01 B), by the Deutsche Forschungsgemeinschaft (DFG) under the ReaSem project, and by the EU in the IST project NeOn (IST-2006-027595, <http://www.neon-project.org/>).

<sup>3</sup> <http://www.w3.org/2004/OWL/>

At the same time, DLP has been a focus of discord in the scientific dispute about the use of open-world versus closed-world knowledge representation and reasoning in the semantic web [12]. We believe, however, that DLP can serve as a basic interoperability layer between these paradigms, at least for scientific investigations, as spelled out in [12]. It may even find more practical uses if considered as a tractable fragment of OWL in the sense of the W3C member submission on OWL 1.1<sup>4</sup>, or as a basis for the W3C Rule Interchange Format RIF<sup>5</sup>, as it provides a bridge e.g. between OWL and the Web Rule Language WRL<sup>6</sup>.

This short technical note has been written with the sole purpose of describing normal forms for DLP, both in Description Logic and in Datalog syntax. We see this as a helpful step for dissemination into adjacent fields of research and possibly also into practice. At the same time, our normal forms can be used as definitions for DLP which – in our opinion – are much more concise and more transparent than others.

For clarification, we note that we do *not* consider Datalog to come with a specific semantics (like the minimal model semantics) which is different from its first-order logic semantics. We simply consider it to be a syntactic fragment of first order logic which thus inherits its semantics. Some people prefer the notion *OWL-Horn* in this case, instead of *DLP*, but it does not really matter in our context.

The paper is structured as follows. In Section 2 we provide normal forms for DLP in both DL and Datalog form, and formally prove that they are indeed normal forms. In Section 3 we give an extended example for DLP using our syntax, and in Section 4 we conclude.

## 2 Normal Forms

We assume that the reader is familiar with basic Description Logics [4], with OWL [3] and basic notions from logic programming [5]. For detailed background on DLP we recommend [2], and for a much shorter overview [1].

We need to fix terminology first. We call DLP the (semantic) fragment common to OWL Lite and Datalog, i.e. we abstract (for the time being) from a concrete syntax: Every OWL Lite statement which is semantically equivalent — in the sense of first order logic — to a (finite) set of function-free Horn clauses (i.e. Datalog rules) constitutes a valid DLP statement. Likewise, every function-free Horn clause which is semantically equivalent to some set of OWL Lite statements constitutes a valid DLP statement<sup>7</sup>. Allowing integrity constraints, we call the

<sup>4</sup> <http://www.w3.org/Submission/2006/10/>

<sup>5</sup> <http://www.w3.org/2005/rules/>

<sup>6</sup> <http://www.w3.org/Submission/2005/08/>

<sup>7</sup> In our terminology, the set of OWL Lite statements  $\{C \sqsubseteq D \sqcup E, D \equiv E\}$  would not qualify as a set of DLP statements, although it is semantically equivalent to  $\{C \sqsubseteq D, D \equiv E\}$ , which is expressible in DLP. We are well aware of this restriction, but will not be concerned with it in the moment, because this more general notion

resulting fragment DLP IC (or just IC). Allowing integrity constraints and equality, we call the resulting fragment DLP ICE (or ICE). We write DLP<sup>+</sup> for the (semantic) fragment common to OWL DL and (function-free non-disjunctive) Datalog. Analogously, we write DLP<sup>+</sup> IC, IC<sup>+</sup>, etc.

In the following, we will give normal forms, both on the Description Logic side and on the Datalog side. I.e. we provide syntactic fragments which allow expressing (semantically) everything in DLP.

## 2.1 Normal Form for Description Logic Syntax

Allowed are the following, where  $a, b, a_i$  stand for individuals,  $C$  stands for a concept name and  $R, Q, R_i, Q_{i,j}$  stand for role names.

- ABox:
  - $C(a)$  (individual assertion)
  - $R(a, b)$  (property assertion)
  - $a = b$  (ICE) (individual equivalence)
- Property Characteristics:
  - $R \equiv Q$  (equivalence)
  - $R \sqsubseteq Q$  (subproperty)
  - $\top \sqsubseteq \forall R.C$  ( $C \neq \perp$ ) (domain)
  - $\top \sqsubseteq \forall R^-.C$  ( $C \neq \perp$ ) (range)
  - $R \equiv Q^-$  (inverse)
  - $R \equiv R^-$  (symmetry)
  - $\top \sqsubseteq \leq 1R$  (ICE) (functionality)
  - $\top \sqsubseteq \leq 1R^-$  (ICE) (inverseFunctionality)
- TBox: We allow expressions of the form

$$\exists Q_{1,1}^{(-)} \dots \exists Q_{1,m_1}^{(-)} \mathbf{Left}_1 \sqcap \dots \sqcap \exists Q_{k,1}^{(-)} \dots \exists Q_{k,m_k}^{(-)} \mathbf{Left}_k \sqsubseteq \forall R_1^{(-)} \dots \forall R_n^{(-)} \mathbf{Right}$$

where the following apply.

- For DLP we allow  $\mathbf{Left}_j$  to be of the forms  $C, \{o_1, \dots, o_n\}, \perp$  or  $\top$ , and  $\mathbf{Right}$  to be of the forms  $C$  or  $\top$ .
- For DLP IC we allow  $\mathbf{Left}_j$  to be of the forms  $C, \{o_1, \dots, o_n\}, \perp$ , or  $\top$ , and  $\mathbf{Right}$  to be of the form  $C, \top$ , or  $\perp$ .
- For DLP ICE we allow  $\mathbf{Left}_j$  to be of the forms  $C, \{o_1, \dots, o_n\}, \perp$ , or  $\top$ , and  $\mathbf{Right}$  to be of the form  $C, \top, \perp$ , or  $\{o\}$ .
- For the DLP<sup>+</sup> versions we furthermore allow  $\mathbf{Right}$  to be of the form  $\exists R^{(-)}. \{a\}$ .

The superscript <sup>(-)</sup> shall indicate, that an inverse symbol may occur in these places. Note that (by a common abuse of notation) we allow any of  $k, m_i, n$  to be zero. For  $k = 0$  the left hand side becomes  $\top$ . Note also that we could have disallowed  $\perp$  on the left and  $\top$  on the right, since in either

---

of semantic equivalence is not readily accessible by syntactic means. Note, however, that  $C \sqsubseteq D \sqcup D$  qualifies as a DLP statement, since it is semantically equivalent to  $C \sqsubseteq D$ .

case the statement becomes void. Likewise, it would suffice to require  $n = 0$  in all cases, since universal quantifiers on the right are expressible using existentials on the left. Disallowing the existential quantifiers on the left (while keeping universals on the right) is also possible, but at the expense of the introduction of an abundance of new concept names. As an example, note that  $\exists R.C \sqcap \exists Q.D \sqsubseteq E$  would have to be translated into the set of statements  $\{C_1 \sqcap D_1 \sqsubseteq E, C \sqsubseteq \forall R^-.C_1, D \sqsubseteq \forall Q^-.D_1\}$ , where  $C_1$  and  $D_1$  are new concept names. Our representation is more compact.

## 2.2 Normal Form for Datalog Syntax

Allowed are the following, where  $x, y, z, y_i, x_{i,j}$  are variables,  $a, b, c, a_j$  are constant symbols,  $C, D$  are unary predicate symbols, and  $Q, R, R_{i,j}$  are binary predicate symbols.

- Corresponding to ABox:
  - $C(a) \leftarrow$  (individual assertion)
  - $R(a, b) \leftarrow$  (property assertion)
  - $a = b \leftarrow$  (individual equivalence)
- Corresponding to Property Characteristics:
  - $Q(x, y) \leftarrow R(x, y)$  (subproperty)
  - $C(y) \leftarrow R(x, y)$  (domain)
  - $C(y) \leftarrow R(y, x)$  (range)
  - $R(x, y) \leftarrow Q(y, x)$  (inverse subproperty)
  - $R(x, y) \leftarrow R(y, x)$  (symmetry)
  - $y = z \leftarrow R(x, y) \wedge R(x, z)$  (ICE) (functionality)
  - $y = z \leftarrow R(y, x) \wedge R(z, x)$  (ICE) (inverseFunctionality)
- Corresponding to TBox: We allow rules of the form

$$\begin{aligned} \text{Left}(y) \leftarrow & Q_{1,1}^{(-)}(x_{1,1}, x_{1,2}) \wedge \cdots \wedge Q_{1,m_1}^{(-)}(x_{1,m_1}, x) \wedge \text{Right}_1(x) \\ & \wedge \dots \\ & \wedge Q_{k,1}^{(-)}(x_{k,1}, x_{k,2}) \wedge \cdots \wedge Q_{k,m_k}^{(-)}(x_{k,m_k}, x) \wedge \text{Right}_k(x) \\ & \wedge R_1^{(-)}(x, y_1) \wedge \cdots \wedge R_n^{(-)}(y_{n-1}, y), \end{aligned}$$

where  $\text{Right}_j(x)$  is of the form  $C(x)$  or  $R^{(-)}(x, a)$ , and  $\text{Left}(y)$  is of the form  $D(y)$ , or (for DLP IC)  $\perp$ , or (for DLP ICE)  $y = b$ , or (for DLP<sup>+</sup> versions)  $Q(y, c)$ . Furthermore, we require all variables  $x, y, y_i, x_{i,j}$  to be mutually distinct.

The meaning of the inverse symbol here is as follows: For a binary predicate symbol  $R$  we let  $R^-(x, y)$  stand for  $R(y, x)$ . A bracketed inverse symbol in the superscript  $(-)$  hence means that the order of the arguments of the corresponding predicate symbol is not relevant.

By slight abuse of notation we allow any of  $k, n, m_j$  to be zero, which may cause the body of the rule to be empty. For  $m_j = 0$  the form of

$$Q_{j,1}^{(-)}(x_{j,1}, x_{j,2}) \wedge \cdots \wedge Q_{j,m_j}^{(-)}(x_{j,m_j}, x) \wedge \text{Right}_j(x)$$

reduces to  $\mathbf{Right}_j(x)$ , with  $\mathbf{Right}_j(x)$  as indicated. For  $n = 0$  we require  $y$  to be  $x$ .

Concerning the terminology just introduced, we can show the following theorem.

**Theorem 1.** *Every  $DLP^{(+)}$  ( $DLP^{(+)}$  IC,  $DLP^{(+)}$  ICE) statement made in normal form for Description logic syntax is semantically equivalent to a set of  $DLP^{(+)}$  ( $DLP^{(+)}$  IC,  $DLP^{(+)}$  ICE) statements made in normal form for Datalog syntax. Conversely, every  $DLP^{(+)}$  ( $DLP^{(+)}$  IC,  $DLP^{(+)}$  ICE) statement made in normal form for Datalog syntax is semantically equivalent to a set of  $DLP^{(+)}$  ( $DLP^{(+)}$  IC,  $DLP^{(+)}$  ICE) statements made in normal form for Description Logic syntax.*

*Proof.* We use the translations between Description Logic and Datalog as provided in [1, 2], and summarized in Table 1. How to obtain the semantically equivalent statements for the ABox and the Property Characteristics parts is evident from this summary.

Now consider a rule

$$\begin{aligned} \mathbf{Left}(y) \leftarrow & Q_{1,1}^{(-)}(x_{1,1}, x_{1,2}) \wedge \cdots \wedge Q_{1,m_1}^{(-)}(x_{1,m_1}, x) \wedge \mathbf{Right}_1(x) \\ & \wedge \dots \\ & \wedge Q_{k,1}^{(-)}(x_{k,1}, x_{k,2}) \wedge \cdots \wedge Q_{k,m_k}^{(-)}(x_{k,m_k}, x) \wedge \mathbf{Right}_k(x) \\ & \wedge R_1^{(-)}(x, y_1) \wedge \cdots \wedge R_n^{(-)}(y_{n-1}, y), \end{aligned}$$

where  $\mathbf{Left}(y)$  and  $\mathbf{Right}_j(x)$  are as indicated above. This translates to the statement

$$\exists Q_{1,1}^{(-)} \dots \exists Q_{1,m_1}^{(-)} \mathbf{Ri}_1 \sqcap \cdots \sqcap \exists Q_{k,1}^{(-)} \dots \exists Q_{k,m_k}^{(-)} \mathbf{Ri}_k \sqsubseteq \forall R_1^{(-)} \dots \forall R_n^{(-)}. \mathbf{Le},$$

where  $\mathbf{Le}$  is

- $D$  if  $\mathbf{Left}(x)$  is  $D(x)$ ,
- $\perp$  if  $\mathbf{Left}(x)$  is  $\perp$ ,
- $\{b\}$  if  $\mathbf{Left}(x)$  is  $x = b$ , and
- $\exists Q^{(-)}. \{c\}$  if  $\mathbf{Left}(x)$  is  $Q^-(x, c)$

and  $\mathbf{Re}_j$  is

- $C$  if  $\mathbf{Right}_j(x_{j,1})$  is  $C(x_{j,1})$ , and
- $\exists R^{(-)}. \{a_j\}$  if  $\mathbf{Right}_j(x_{j,1})$  is  $R^{(-)}(x_{j,q}, a_j)$ .

We need to justify our translation by showing that the resulting Datalog rule is semantically equivalent to the Description Logic statement from which it was obtained. It boils down to somewhat tedious equivalence transformations in first order logic following the exhibitions in [1, 2], and we will not be bothered with

OWL DL	DL statement		DLP rule
<b>ABox</b>			
indiv. assertion	$C(a)$		$C(a) \leftarrow$
property assertion	$R(a, b)$		$R(a, b) \leftarrow$
indiv. equiv.	$a = b$	ICE	$a = b \leftarrow$
indiv. inequiv.	$\neg(a = b)$		not expressible in general
<b>TBox</b>			
equivalence	$C \equiv D$		not expressible in general
GCI	$C \sqsubseteq D$		$D(x) \leftarrow C(x)$
top	$\top$		expressible
bottom	$C \sqsubseteq \perp$	IC (ri)	$\perp \leftarrow C(x)$
conjunction	$C \sqcap D \sqsubseteq E$		$E(x) \leftarrow C(x) \wedge D(x)$
	$C \sqsubseteq E \sqcap F$		$E(x) \leftarrow C(x)$ $F(x) \leftarrow C(x)$
disjunction	$C \sqcup D \sqsubseteq E$	(le)	$E(x) \leftarrow C(x)$ $E(x) \leftarrow D(x)$
			not expressible in general
atomic negation	$\neg A$		not expressible in general
univ. restriction	$D \sqsubseteq \forall R.C$ ( $C \neq \perp$ )	(ri)	$C(y) \leftarrow D(x) \wedge R(x, y)$
	$D \sqsubseteq \forall R.\perp$	IC (ri)	$\perp \leftarrow D(x) \wedge R(x, y)$
exist. restriction	$\exists R.C \sqsubseteq D$ ( $C \neq \perp$ )	(le)	$D(x) \leftarrow R(x, y) \wedge C(y)$
	$\exists R.\perp \sqsubseteq D$	IC (le)	$\perp \leftarrow R(x, y) \wedge C(y)$
one-of 1	$C \sqsubseteq \{a\}$	ICE	$a = x \leftarrow C(x)$
hasValue	$\{a\} \sqsubseteq C$		$C(a) \leftarrow$
	$\exists R.\{a\} \sqsubseteq C$		$C(x) \leftarrow R(x, a)$
one-of	$C \sqsubseteq \exists R.\{a\}$	DLP <sup>+</sup>	$R(x, a) \leftarrow C(x)$
	$\{o_1, \dots, o_n\} \sqsubseteq C$	(le)	$C(o_i) \leftarrow$ (for $i = 1, \dots, n$ )
card. restrictions	...		not expressible in general
<b>Property Characteristics</b>			
equivalence	$R \equiv Q$		$R(x, y) \leftarrow Q(x, y)$ $Q(x, y) \leftarrow R(x, y)$
subproperty	$R \sqsubseteq Q$		$Q(x, y) \leftarrow R(x, y)$
domain	$\top \sqsubseteq \forall R.C$ ( $C \neq \perp$ )		$C(y) \leftarrow R(x, y)$
range	$\top \sqsubseteq \forall R^-.C$ ( $C \neq \perp$ )		$C(y) \leftarrow R(y, x)$
inverse	$R \equiv Q^-$		$R(x, y) \leftarrow Q(y, x)$ $Q(x, y) \leftarrow R(y, x)$
symmetry	$R \equiv R^-$		$R(x, y) \leftarrow R(y, x)$
transitivity			$R(x, y) \leftarrow R(x, z) \wedge R(z, y)$
functionality	$\top \sqsubseteq \leq 1R$	ICE	$y = z \leftarrow R(x, y) \wedge R(x, z)$
inverseFunctionality	$\top \sqsubseteq \leq 1R^-$	ICE	$y = z \leftarrow R(y, x) \wedge R(z, x)$

**Table 1.** Translation from DL to Datalog, taken from [1, 2]. The abbreviation *ri* (*le*) means *right* (*left*) of GCI only.

the details. We can, however, make our transformation transparent by means of the transformations listed in Table 1. The statement

$$\exists Q_{1,1}^{(-)} \dots \exists Q_{1,m_1}^{(-)} \mathbf{Ri}_1 \sqcap \dots \sqcap \exists Q_{k,1}^{(-)} \dots \exists Q_{k,m_k}^{(-)} \mathbf{Ri}_k \sqsubseteq \forall R_1^{(-)} \dots \forall R_n^{(-)}. \mathbf{Le}$$

can be written as the pair of statements

$$\begin{aligned} \exists Q_{1,1}^{(-)} \dots \exists Q_{1,m_1}^{(-)} \mathbf{Ri}_1 \sqcap \dots \sqcap \exists Q_{k,1}^{(-)} \dots \exists Q_{k,m_k}^{(-)} \mathbf{Ri}_k &\sqsubseteq D \\ D &\sqsubseteq \forall R_1^{(-)} \dots \forall R_n^{(-)}. \mathbf{Le}, \end{aligned}$$

where  $D$  is a new concept name. These statements can be translated separately into

$$\begin{aligned} D(x) \leftarrow & Q_{1,1}^{(-)}(x_{1,1}, x_{1,2}) \wedge \dots \wedge Q_{1,m_1}^{(-)}(x_{1,m_1}, x) \wedge \mathbf{Right}_1(x) \\ & \wedge \dots \\ & \wedge Q_{k,1}^{(-)}(x_{k,1}, x_{k,2}) \wedge \dots \wedge Q_{k,m_k}^{(-)}(x_{k,m_k}, x) \wedge \mathbf{Right}_k(x) \end{aligned}$$

and

$$\mathbf{Left}(y) \leftarrow D(x) \wedge R_1^{(-)}(x, y_1) \wedge \dots \wedge R_n^{(-)}(y_{n-1}, y).$$

By unfolding over  $D(x)$  we obtain the desired combined rule.

The translation can obviously be performed in both directions, so there is nothing more to show.

It is possible to strengthen Theorem 1 by providing a translation between single Description Logic statements and single Datalog rules (in normal form). In this case we would have to disallow the property characteristics *inverse* on the OWL side, which can be done since  $R \equiv Q^-$  is expressible e.g. by the set of statements  $\{R \sqsubseteq Q^-, Q^- \sqsubseteq R\}$ , each member of which is in turn translatable into a single Datalog statement. Similarly, property equivalence would have to be disallowed. We think that the form we have chosen is more concise.

**Theorem 2.** *All description logic programs following [1, 2] can be written in normal form.*

*Proof.* All statements belonging to DLP as described in [1, 2] are listed in Table 1. It is easy to check that all possibly resulting Datalog statements listed in the last column are already in normal form, which suffices to show the statement.

### 3 Examples

A rule of thumb for the creation of DLP ontologies is: *Avoid concrete domains and number restrictions, and be careful with quantifiers, disjunction, and nominals.* We give a small example ontology which includes the safe usage of the latter constructs. It shall display the modelling expressivity of DLP.

For the TBox, we model the following sentences.

- (1) Every man or woman is an adult.
- (2) A grown-up is a human who is an adult.
- (3) A woman who has somebody as a child, is a mother.
- (4) An orphan is the child of humans who are dead.
- (5) A lonely child has no siblings.
- (6) AIFB researchers are employed by the University of Karlsruhe.

They can be written in DLP as follows.

- $$\begin{aligned} \text{Man} \sqcup \text{Woman} &\sqsubseteq \text{Adult} & (1) \\ \text{GrownUp} &\sqsubseteq \text{Human} \sqcap \text{Adult} & (2) \\ \text{Woman} \sqcap \exists \text{childOf}^- . \top &\sqsubseteq \text{Mother} & (3) \\ \text{Orphan} &\sqsubseteq \forall \text{childOf}. (\text{Dead} \sqcap \text{Human}) & (4) \\ \text{LonelyChild} &\sqsubseteq \neg \exists \text{siblingOf}. \top & (5) \\ \text{AIFBResearcher} &\sqsubseteq \exists \text{employedBy}. \{\text{UKARL}\} & (6) \end{aligned}$$

In normal form in Description Logic syntax these are as follows.

- $$\begin{aligned} \text{Man} &\sqsubseteq \text{Adult} & (1) \\ \text{Woman} &\sqsubseteq \text{Adult} & (1) \\ \text{GrownUp} &\sqsubseteq \text{Human} & (2) \\ \text{GrownUp} &\sqsubseteq \text{Adult} & (2) \\ \text{Woman} \sqcap \exists \text{childOf}^- . \top &\sqsubseteq \text{Mother} & (3) \\ \text{Orphan} &\sqsubseteq \forall \text{childOf}. \text{Dead} & (4) \\ \text{Orphan} &\sqsubseteq \forall \text{childOf}. \text{Human} & (4) \\ \text{LonelyChild} &\sqsubseteq \forall \text{siblingOf}. \perp & (5) \\ \text{AIFBResearcher} &\sqsubseteq \exists \text{employedBy}. \{\text{UKARL}\} & (6) \end{aligned}$$

We note that for (5) we require DLP IC, while for (6) we require DLP<sup>+</sup>. For the RBox, we use the following.

- |  |   |
|--|---|
| $\text{parentOf} \equiv \text{childOf}^-$                  | parentOf and childOf are inverse roles. |
| $\text{parentOf} \sqsubseteq \text{ancestorOf}$            | parentOf is a subrole of ancestorOf.    |
| $\text{fatherOf} \sqsubseteq \text{parentOf}$              | fatherOf is a subrole of parentOf.      |
| $\top \sqsubseteq \forall \text{ancestorOf}. \text{Human}$ | Human is the domain of ancestorOf.      |
| $\top \sqsubseteq \leq 1 \text{fatherOf}^-$                | fatherOf is inverse functional.         |

We can populate the classes and roles by means of an ABox in the following way.

- $$\begin{aligned} \{\text{Bernhard}, \text{Benedikt}, \text{Rainer}, \text{Ganter}\} &\sqsubseteq \text{Man} \\ \{\text{Ruth}, \text{Ulrike}\} &\sqsubseteq \text{Woman} \\ \text{Bernhard} &= \text{Ganter} \\ \text{employedBy}(\text{Bernhard}, \text{TUD}) & \\ \dots & \end{aligned}$$



Note that an ABox statement such as

$$\{\text{Ruth}, \text{Ulrike}\} \sqsubseteq \text{Woman}$$

is simply syntactic sugar for the two statements

$$\text{Woman}(\text{Ruth}) \quad \text{Woman}(\text{Ulrike})$$

We therefore consider it to be part of the ABox. To be precise, the original statement is (syntactically) not in OWL Lite, but the equivalent set of three ABox statements is. The statement  $\text{Bernhard} = \text{Ganter}$  requires DLP ICE.

Note also that class inclusions cannot in general be replaced by equivalences. For example, the statement

$$\text{Adult} \sqsubseteq \text{Man} \sqcup \text{Woman}$$

is not in DLP.

For illustration, we give the knowledge base in Datalog normal form. The TBox is as follows.

$$\text{Adult}(y) \leftarrow \text{Man}(y) \tag{1}$$

$$\text{Adult}(y) \leftarrow \text{Woman}(y) \tag{1}$$

$$\text{Human}(y) \leftarrow \text{GrownUp}(y) \tag{2}$$

$$\text{Adult}(y) \leftarrow \text{GrownUp}(y) \tag{2}$$

$$\text{Mother}(y) \leftarrow \text{childOf}(x, y) \wedge \text{Woman}(y) \tag{3}$$

$$\text{Dead}(y) \leftarrow \text{Orphan}(x) \wedge \text{childOf}(x, y) \tag{4}$$

$$\text{Human}(y) \leftarrow \text{Orphan}(x) \wedge \text{childOf}(x, y) \tag{4}$$

$$\leftarrow \text{LonelyChild}(x) \wedge \text{siblingOf}(x, y) \tag{5}$$

$$y = \text{UKARL} \leftarrow \text{AIFBResearcher}(x) \wedge \text{employedBy}(x, y) \tag{6}$$

Translating the RBox yields the following statements.

$$\text{parentOf}(x, y) \leftarrow \text{childOf}(y, x)$$

$$\text{childOf}(x, y) \leftarrow \text{childOf}(y, x)$$

$$\text{ancestorOf}(x, y) \leftarrow \text{parentOf}(x, y)$$

$$\text{parentOf}(x, y) \leftarrow \text{fatherOf}(x, y)$$

$$\text{Human}(y) \leftarrow \text{ancestorOf}(x, y)$$

$$y = z \leftarrow \text{fatherOf}(y, x) \wedge \text{fatherOf}(z, x)$$

## 4 Conclusions

We have presented normal forms for Description Logic Programs, both in Description Logic syntax and in Logic Programming syntax. We have formally shown that these are indeed normal forms.

We believe that these normal forms can and should be used for *defining* Description Logic Programs. We have found that some of the definitions used in the literature remain somewhat ambiguous, so that the language is not entirely specified. This brief note rectifies this problem in providing a frame of reference.

## References

1. Grosz, B., Horrocks, I., Volz, R., Decker, S.: Description logic programs: Combining logic programs with description logics. In: Proc. of WWW 2003, Budapest, Hungary, May 2003, ACM (2003) 48–57
2. Volz, R.: Web Ontology Reasoning with Logic Databases. PhD thesis, University of Manchester (2004)
3. Antoniou, G., van Harmelen, F.: Web Ontology Language: OWL. In Staab, S., Studer, R., eds.: Handbook on Ontologies. Springer (2004) 67–92
4. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
5. Lloyd, J.W.: Foundations of Logic Programming. Springer, Berlin (1988)
6. Angele, J., Lausen, G.: Ontologies in F-logic. In Staab, S., Studer, R., eds.: Handbook on Ontologies. Springer (2004) 29–50
7. Hustadt, U., Motik, B., Sattler, U.: Reducing SHIQ-description logic to disjunctive datalog programs. In Dubois, D., Welty, C.A., Williams, M.A., eds.: Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004), Whistler, Canada, June 2-5, 2004, AAAI Press (2004) 152–162
8. Hustadt, U., Motik, B., Sattler, U.: Reasoning in description logics with a concrete domain in the framework of resolution. In de Mántaras, R.L., Saitta, L., eds.: Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004, IOS Press (2004) 353–357
9. Motik, B.: Reasoning in description logics using resolution and deductive databases. PhD thesis, University Karlsruhe, Germany (2006)
10. Motik, B., Sattler, U., Studer, R.: Query answering for OWL-DL with rules. Journal of Web Semantics: Science, Services and Agents on the World Wide Web **3** (2005) 41–60
11. de Bruijn, J., Polleres, A., Lara, R., Fensel, D.: OWL<sup>-</sup>. WSML Working Draft, DERI (2004)
12. Hitzler, P., Haase, P., Krötzsch, M., Sure, Y., Studer, R.: DLP isn't so bad after all. In Grau, B.C., Horrocks, I., Parsia, B., Patel-Schneider, P., eds.: Proceedings of the Workshop OWL – Experiences and Directions, Galway, Ireland. Volume 188 of CEUR Workshop Proceedings. (2005) Online <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-188/sub2.pdf>.