Translating Ontologies from Predicate-based to Frame-based Languages

Jos de Bruijn and Stijn Heymans

Digital Enterprise Research Institute (DERI)
University of Innsbruck, Austria
{jos.debruijn,stijn.heymans}@deri.org

Abstract

Many popular ontology languages are based on (subsets of) first-order predicate logic, where classes are modeled as unary predicates and properties as binary predicates, such as Description Logics. Specifically, the ontology language OWL DL is based on the Description Logic $\mathcal{SHOIQ}$. F-Logic is a different ontology language which is also based on first-order logic, but classes and properties are represented as terms, rather than predicates. In this paper we define a translation from predicate-based ontologies to F-Logic ontologies and show that this translation preserves entailments for large classes of ontologies, including most of OWL DL. We define the class of $\mathcal{E}$-safe formulas, show that the Description Logic $\mathcal{SHIQ}$ is $\mathcal{E}$-safe, and show that the translation preserves validity of $\mathcal{E}$-safe formulas. We then use these results to close the open problems of layering F-Logic programming on top of Description Logic Programs and language layering in WSML.

1 Introduction

There have been several proposals for using F-Logic as the basis for an ontology language for the Semantic Web [17, 10, 2, 6]. In F-Logic, classes and properties are interpreted as objects. This may hamper inter-operation with Description Logic-based ontology languages (e.g. OWL DL [12]), in which classes and properties are interpreted as unary and binary predicates. We will call the way of modeling ontologies in F-Logic "frame-based ontology modeling" and the way of modeling ontologies in Description Logics "predicate-based ontology modeling".

More specifically, SWSL [6], WRL [2] and WSML [10] claim that an F-Logic based variant of the language (WRL- resp. WSML-Flight) is an extension of a Description Logic (Programming) based variant of the language (WRL- resp. WSML-Core). It is an open problem whether the F-Logic based variants are really extensions of the Description Logic based variant.

We define a translation from predicate-based ontologies to F-Logic. We show that when considering sorted F-Logic, the translation preserves entailment for arbitrary first-order theories. We then show that this is not the case in general when translating the ontology to an unsorted F-Logic language, but for certain classes of first-order formulas, called the *cardinal* formulas, the entailments are equivalent. Our translation preserves function-freeness, i.e., if no function symbol of arity > 0 was used in the original ontology, no function symbol of arity > 0 will occur in the translated ontology.

We show that the translation to unsorted F-Logic preserves validity for large classes of predicate-based languages. We define the class of $\mathcal{E}$-safe formulas, show that the Description Logic $\mathcal{SHIQ}$ is $\mathcal{E}$-safe, and show that $\mathcal{E}$-safe formulas are cardinal. Finally, $\mathcal{E}$-safe formulas are closed under negation, and thus entailment of $\mathcal{E}$-safe formulas can be reduced to checking validity.

Our definition of cardinal formulas originates from HiLog [7], and thus the class of cardinal formulas we describe in this paper, the $\mathcal{E}$-safe formulas, can be used for HiLog as well.

We use these results to close the open problem of F-Logic extensions of Description Logic Programs [14] and the problem of language layering in WSML (and thus also WRL). We show that the WSML variants are indeed semantically layered as suggested in [10]. Specifically, we show that the language layering preserves (ground) entailment

Structure of the paper In Section 2 we review predicate- and frame-based ontology modeling languages. In Section 3, We show that the translation of any predicate-based ontology to sorted F-Logic is faithful and that the translation of *cardinal* formulas to unsorted F-Logic is faithful; we identify the class of $\mathcal{E}$-safe formulas and demonstrate cardinality. We use this translation to show that the straightforward F-Logic extension of DLP preserves ground entailment, in Section 4. We then use the translation to show that the WSML language variant are layered, in Section 5. Finally, we review related work in Section 6 and present conclusions in Section 7.

2 Preliminaries

Predicate-based ontology modeling A predicate-based ontology language is a first-order language in which unary predicates represent classes of objects and binary predicates represent properties (relations between objects). Description Logics [3] are such predicate-based ontology languages. Of special interest is $\mathcal{SHOIQ}$, which is the language underlying OWL DL. We present the syntax and semantics of $\mathcal{SHOIQ}$ through a mapping to first-order logic with equality. The descriptions are presented in Table 1; the axioms are presented in Table 2. In the tables, A is a named class, C, D are descriptions, Q, R are roles, and $a, b, o_1, ..., o_n$ are individuals. Additionally, we have that in the *number restrictions* $\geqslant nR.C$ and $\leqslant nR.C$, R have to be *simple*, i.e., R and its subroles may not be transitive (with transitivity indicated by $\mathsf{Trans}(R)$)

The Description Logic $\mathcal{SHIQ}$ corresponds to $\mathcal{SHOIQ}$ without the enumeration ($\{o_1, ..., o_n\}$) and hasvalue ($\exists R.\{o\}$) descriptions. In the remainder of the paper, when referring to $\mathcal{SHOIQ}$ (resp. $\mathcal{SHIQ}$) axioms, we refer to the FOL version of these axioms.

DL syntax	FOL syntax
$\pi_y(A, X)$	$A(X)$
$\pi_y(\top, X)$	$X = X$
$\pi_y(\bot, X)$	$\neg(X = X)$
$\pi_y(C_1 \sqcap \ldots \sqcap C_n, X)$	$\bigwedge \pi_y(C_i, X)$
$\pi_y(C_1 \sqcup \ldots \sqcup C_n, X)$	$\bigvee \pi_y(C_i, X)$
$\pi_y(\neg C, X)$	$\neg\pi_y(C, X)$
$\pi_y(\{o_1 \ldots o_n\}, X)$	$\bigvee X = o_i$
$\pi_y(\exists R.C, X)$	$\exists y(R(X, y) \wedge \pi_x(C, y))$
$\pi_y(\forall R.C, X)$	$\forall y(R(X, y) \supset \pi_x(C, y))$
$\pi_y(\exists R.\{o\}, X)$	$R(X, o)$
$\pi_y(\geqslant nR.C, X)$	$\exists y_1, \ldots, y_n(\bigwedge R(X, y_i) \wedge \bigwedge \pi_x(C, y_i) \wedge \bigwedge \neg y_i = y_j)$
$\pi_y(\leqslant nR.C, X)$	$\forall y_1, \ldots, y_{n+1}(\bigwedge R(X, y_i) \wedge \bigwedge \pi_x(C, y_i) \supset \bigvee y_i = y_j)$
π_x is defined as π_y by substituting x and x_i for all y and y_i, respectively	

Table 1. $\mathcal{SHOIQ}$ **Descriptions**

We distinguish between concept (unary predicate) and role (binary predicate) symbols, and predicate symbols of other arities, in the signature of the language. A first-order signature Σ has the form $\Sigma = \langle \mathcal{A}, \mathcal{C}, \mathcal{R}, \mathcal{P} \rangle$, where $\mathcal{A}$ is the set of function symbols, each with an associated arity n, $\mathcal{C}$ is a set of concept (unary predicate) symbols, $\mathcal{R}$ is a set of role (binary predicate) symbols, and $\mathcal{P}$ is a set of n-ary predicate symbols, with $n = 0$ or $n \geq 3$. $\mathcal{A}, \mathcal{C}, \mathcal{R}$, and $\mathcal{P}$ are disjoint.

DL syntax	FOL syntax
Class Axioms	
$C \sqsubseteq D$	$\forall x(\pi_y(C, x) \supset \pi_y(D, x))$
$C \equiv D$	$\forall x(\pi_y(C, x) \supset \pi_y(D, x)) \wedge \forall x(\pi_y(D, x) \supset \pi_y(C, x))$
Property Axioms	
$Q \sqsubseteq R$	$\forall x, y(Q(x, y) \supset R(x, y))$
$R \equiv Q^-$	$\forall x, y(R(x, y) \supset Q(y, x)) \wedge \forall x, y(Q(y, x) \supset R(x, y))$
$\mathsf{Trans}(R)$	$\forall x, y, z(R(x, y) \wedge R(y, z) \supset R(x, z))$
Individual Axioms	
$a \in A$	$A(a)$
$\langle a, b \rangle \in R$	$R(a, b)$
$a = b$	$a = b$
$a \neq b$	$\neg(a = b)$

Table 2. $\mathcal{SHOIQ}$ **Axioms**

Given a signature Σ and a set of variable symbols $\mathcal{V}$, terms are either variables or constructed terms of the form $f(t_1, ..., t_n)$ with $f \in \mathcal{A}$ an n-ary function symbol ($n \geq 0$) and $t_1, ..., t_n$ terms. Atomic formulas are expressions of the form $p(t_1, ..., t_n)$ with $p \in \mathcal{C} \cup \mathcal{R} \cup \mathcal{P}$ an n-ary predicate symbol ($n \geq 0$) and $t_1, ..., t_n$ terms. Formulas of a first-order language $\mathcal{L}^P$ are constructed as usual: every atomic formula is a formula in $\mathcal{L}^P$; compound formulas are constructed using atomic formulas, the logical connectives $\neg, \wedge, \vee, \supset$, the quantifiers $\exists, \forall$ and the auxiliary symbols $), ($.

An *interpretation* for a language $\mathcal{L}^P$ is a tuple $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$, where Δ is a nonempty set (called *domain*) and $\cdot^{\mathcal{I}}$ is a mapping which assigns: a function $f^{\mathcal{I}} : \Delta^n \to \Delta$ to every n-ary function symbol $f \in \mathcal{A}$, and a relation $p^{\mathcal{I}} \subseteq \Delta^n$, to every n-ary predicate symbol $p \in \mathcal{C} \cup \mathcal{R} \cup \mathcal{P}$.

A variable assignment B is a mapping which assigns an element $x^B \in \Delta$ to every variable symbol x. A variable assignment B' is an x-variant of B if $y^B = y^{B'}$ for every variable $y \in \mathcal{V}$ for $y \neq x$.

Given an interpretation $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$, a variable assignment B, and a term t of $\mathcal{L}^P$, $t^{\mathcal{I},B}$ is defined as: $x^{\mathcal{I},B} = x^B$ for variable symbol x and $t^{\mathcal{I},B} = f^{\mathcal{I}}(t_1^{\mathcal{I},B}, ..., t_n^{\mathcal{I},B})$ for t of the form $f(t_1, ..., t_n)$. $\mathcal{I}$ *satisfies* an atomic formula $p(t_1, ..., t_n)$, given a variable assignment B, denoted $\mathcal{I}, B \models p(t_1, ..., t_n)$, if $(t_1^{\mathcal{I},B}, ..., t_n^{\mathcal{I},B}) \in p^{\mathcal{I}}$. $\mathcal{I}, B \models t_1 = t_2$ iff $t_1^{\mathcal{I},B} = t_2^{\mathcal{I},B}$. This is extended to arbitrary formulas as usual: $\mathcal{I}, B \models \phi_1 \wedge \phi_2$ (resp. $\mathcal{I}, B \models \phi_1 \vee \phi_2$, $\mathcal{I}, B \models \neg\phi_1$) iff $\mathcal{I}, B \models \phi_1$ and $\mathcal{I}, B \models \phi_2$ (resp. $\mathcal{I}, B \models \phi_1$ or $\mathcal{I}, B \models \phi_2$, $\mathcal{I}, B \not\models \phi_1$); $\mathcal{I}, B \models \forall x(\phi_1)$ (resp. $\mathcal{I}, B \models \exists x(\phi_1)$) iff for every (resp. for some) B' which is an x-variant of B, $\mathcal{I}, B' \models \phi_1$.

An interpretation $\mathcal{I}$ is a *model* of ϕ, denoted $\mathcal{I} \models \phi$, if $\mathcal{I}, B \models \phi$ for all variable assignments B; ϕ is satisfiable if it has a model (unsatisfiable otherwise); ϕ is valid if every interpretation $\mathcal{I}$ is a model of ϕ. These definitions are straightforwardly extended to the case of first-order theories $\Phi \subseteq \mathcal{L}^P$.

A theory $\Phi \subseteq \mathcal{L}^P$ *entails* a formula $\phi \in \mathcal{L}^P$, denoted $\Phi \models \phi$, iff for all interpretations $\mathcal{I}$ in $\mathcal{L}^P$ such that $\mathcal{I} \models \Phi$, $\mathcal{I} \models \phi$.

Frame-based ontology modeling Frame Logic [17, 18] (F-Logic) is an extension of first-order logic which adds explicit support for object-oriented modeling. It is possible to explicitly specify methods, as well as generalization/specialization and instantiation relationships. The syntax of F-Logic has some seemingly higher-order features, namely, the same identifier can be used for a class, an instance, and a method. However, the semantics of F-Logic is strictly first-order. To simplify matters, we do not consider parametrized methods, functional (single-valued) methods, inheritable methods, and compound molecules.

The signature of an F-Logic language $\mathcal{L}^F$ is of the form $\Sigma = \langle \mathcal{F}, \mathcal{P} \rangle$ with $\mathcal{F}$ a set of function symbols and $\mathcal{P}$ a set of predicate symbols, each with an associated arity $n \geq 0$. Let $\mathcal{V}$ be a set of variable symbols. Terms and atomic formulas are constructed as in first-order logic: $x \in \mathcal{V}$ is a term and $f(t_1, ..., t_n)$ is a term, with $f \in \mathcal{F}$ an n-ary function symbol and $t_1, ..., t_n$ terms.

A molecule in F-Logic is one of the following statements: (i) an *is-a* assertion of the form $C : D$, (ii) a *subclass-of* assertion of the form $C :: D$, or (iii) a data molecule of the form $C[D \rightarrow\!\!\!\rightarrow E]$. with C, D, E terms. An F-Logic molecule is *ground* if it does not contain variables.

Formulas of an F-language $\mathcal{L}^F$ are either atomic formulas, molecules, or compound formula which are constructed in the usual way from atomic formulas, molecules, and the logical connectives $\neg, \wedge, \vee, \supset$, the quantifiers $\exists, \forall$ and the auxiliary symbols $), ($. We denote universal closure with $(\forall)$.

F-Logic Horn formulas are of the form $(\forall) B_1 \wedge ... \wedge B_n \supset H$, with $B_1, ..., B_n, H$ atomic formulas or molecules. F-Logic Datalog formulas are F-Logic Horn formulas without function symbols and where every variable in H occurs in $B_1, ..., B_n$.

Interpretations in F-Logic are called *F-structures*. An *F-structure* is a tuple $\mathbf{I} = \langle U, \prec_U, \in_U, \mathbf{I}_F, \mathbf{I}_P, \mathbf{I}_{\rightarrow\!\!\!\rightarrow} \rangle$. Here, $\prec_U$ is an irreflexive partial order on the domain U and $\in_U$ is a binary relation over U. We write $a \preceq_U b$ when $a \prec_U b$ or $a = b$, for $a, b \in U$. For each F-structure holds that if $a \in_U b$ and $b \preceq_U c$ then $a \in_U c$. Thus, if $b \preceq_U c$, then $\{k \mid k \in_U b, k \in U\} \subseteq \{k \mid k \in_U c, k \in U\}$.

An n-ary function symbol $f \in F$ is interpreted as a function over the domain U: $\mathbf{I}_F(f) : U^n \rightarrow U$. An n-ary predicate symbol $p \in P$ is interpreted as a relation over the domain U: $\mathbf{I}_P(p) \subseteq U^n$. $\mathbf{I}_{\rightarrow\!\!\!\rightarrow}$ associates a partial function $U \rightarrow \mathcal{P}(U)$[1] with each element of U: $\mathbf{I}_{\rightarrow\!\!\!\rightarrow} : U \longrightarrow U \rightarrow \mathcal{P}(U)$.

[1] $\mathcal{P}(U)$ denotes the powerset of U.

Variable assignments are as in first-order logic.

Given an interpretation $\mathbf{I}$, a variable assignment B, and a term t of $\mathcal{L}^F$, $t^{\mathbf{I},B}$ is defined as: $x^{\mathbf{I},B} = x^B$ for variable symbol x and $t^{\mathbf{I},B} = \mathbf{I}_F(f)(t_1^{\mathbf{I},B}, ..., t_n^{\mathbf{I},B})$ for t of the form $f(t_1, ..., t_n)$.

Satisfaction of ϕ in $\mathbf{I}$, given the variable assignment B, denoted $\mathbf{I}, B \models_{\mathsf{f}} \phi$, is defined as: (a) $\mathbf{I}, B \models_{\mathsf{f}} p(t_1, ..., t_n)$ iff $(t_1^{\mathbf{I},B}, ..., t_n^{\mathbf{I},B}) \in \mathbf{I}_P(p)$, (b) $\mathbf{I}, B \models_{\mathsf{f}} t_1 : t_2$ iff $t_1^{\mathbf{I},B} \in_U t_2^{\mathbf{I},B}$, (c) $\mathbf{I}, B \models_{\mathsf{f}} t_1 :: t_2$ iff $t_1^{\mathbf{I},B} \preceq_U t_2^{\mathbf{I},B}$, (d) $\mathbf{I}, B \models_{\mathsf{f}} t_1[t_2 \rightarrow\!\!\!\rightarrow t_3]$ iff $\mathbf{I}_{\rightarrow\!\!\!\rightarrow}(t_2^{\mathbf{I},B})(t_1^{\mathbf{I},B})$ is defined and $t_3^{\mathbf{I},B} \in \mathbf{I}_{\rightarrow\!\!\!\rightarrow}(t_2^{\mathbf{I},B})(t_1^{\mathbf{I},B})$, and (e) $\mathbf{I}, B \models_{\mathsf{f}} t_1 = t_2$ iff $t_1^{\mathbf{I},B} = t_2^{\mathbf{I},B}$. Extension to satisfaction of compound formulas is as in first-order logic.

The notions of a model and of validity are defined analogous to first-order logic. A theory $\Phi \subseteq \mathcal{L}^F$ *F-entails* a formula $\phi \in \mathcal{L}^F$, denoted $\Phi \models_{\mathsf{f}} \phi$, iff for all F-structures $\mathbf{I}$ such that $\mathbf{I} \models_{\mathsf{f}} \Phi$, $\mathbf{I} \models_{\mathsf{f}} \phi$.

Sorted F-Logic In predicate-based ontology modeling, the sets of symbols used for concepts, roles and individuals are disjoint. This is not the case in F-Logic. This disjointness can be regained by using a *sorted* F-Logic language.

We consider a sorted F-Logic language with three sorts: individuals, concepts and roles. A sorted F-Logic language has a sorted signature $\Sigma = \langle \mathcal{A}, \mathcal{C}, \mathcal{R}, \mathcal{P} \rangle$, where $\mathcal{A}$ is the set of function symbols, $\mathcal{C}$ is a set of concept (nullary function) symbols, $\mathcal{R}$ is a set of role (nullary function) symbols, and $\mathcal{P}$ is a set of n-ary predicate symbols, with $n \geq 0$. $\mathcal{A}, \mathcal{C}, \mathcal{R}$, and $\mathcal{P}$ are disjoint. The usual restrictions to the use of symbols in formulas applies, namely only molecules of the form $a : c, c :: d, a[r \rightarrow\!\!\!\rightarrow b]$ are allowed, with a, b terms constructed from symbols in $\mathcal{A}, \mathcal{V}$, $c, d \in \mathcal{C} \cup \mathcal{V}$, and $r \in \mathcal{R} \cup \mathcal{V}$. Quantifiers need to be qualified with i, c, r to indicate over which domain (individual, concept, role) the variable quantifies.

A sorted F-structure has three disjoint domains: U_i, U_c, U_r for the individuals, concepts, and roles, respectively; $\prec_U$ is an irreflexive partial order over U_c; $\in_U$ is a relation between U_i and U_c: $\in_U$: $U_i \times U_c$. $\mathbf{I}_F$ interprets symbols in $\mathcal{A}$ as functions over U_i, symbols in $\mathcal{C}$ as elements in U_c, and symbols of $\mathcal{R}$ as elements in U_r. $\mathbf{I}_P$ interprets symbols in $\mathcal{P}$ as n-ary relations over U_i^n. Finally, $\mathbf{I}_{\rightarrow\!\!\!\rightarrow}$ associates a partial mapping $U_i \rightarrow \mathcal{P}(U_i)$ to each element of U_r.

3 Translating Predicate-Based Ontologies to F-Logic

Table 3 defines a mapping from the predicate style of ontology modeling to the frame style. In the table, A, B are unary predicate symbols, C, D are formulas, R is a binary

Entity	Predicate style	Frame style
Class	$\delta(A(X))$	$X : A$
Property value	$\delta(R(X,Y))$	$X[R \twoheadrightarrow Y]$
Equality	$\delta(X = Y)$	$X = Y$
n-ary relation	$\delta(P(\vec{X}))$	$P(\vec{X})$
Universal	$\delta(\forall\vec{x}.C)$	$\forall\vec{x}.\delta(C)$
Existential	$\delta(\exists\vec{x}.C)$	$\exists\vec{x}.\delta(C)$
Conjunction	$\delta(C \wedge D)$	$(\delta(C) \wedge \delta(D))$
Disjunction	$\delta(C \vee D)$	$(\delta(C) \vee \delta(D))$
Implication	$\delta(C \supset D)$	$(\delta(C) \supset \delta(D))$
Negation	$\delta(\neg C)$	$\neg(\delta(C))$

Table 3. Translating predicate-based to frame-based modeling

predicate symbol, P is an n-ary relation symbol, with $n = 0$ or $n \geq 3$, x, y, z are variable symbols, a, b are constant symbols, and X, Y are terms. The mapping δ is extended to sets of formulas in the usual way.

Definition 1 (Translating formulas)**.** *Given a predicate-based ontology language $\mathcal{L}^P$ with the signature $\Sigma_{\mathcal{L}^P} = \langle \mathcal{A}, \mathcal{C}, \mathcal{R}, \mathcal{P} \rangle$. Let $\mathcal{L}^F$ be the corresponding F-Logic language which has the signature $\Sigma_{\mathcal{L}^F} = \langle \mathcal{F}, \mathcal{P} \rangle$, with $\mathcal{F} = \mathcal{A} \cup \mathcal{C} \cup \mathcal{R}$.*

Given a set of first-order formulas $\Phi \subseteq \mathcal{L}^P$, then $\delta(\Phi) \subseteq \mathcal{L}^F$ is the corresponding set *of F-Logic formulas, with δ as in Table 3.*

In the remainder of this section, we will first show that the translation in Definition 1 is faithful (i.e., preserves entailment) when considering a sorted F-Logic language. We will then show that for a certain class of formulas, the class of *cardinal* formulas, the translation is also faithful when considering an unsorted language. Besides the classes of cardinal formulas identified in [7], we identify the class of $\mathcal{E}$-safe formulas, show that reasoning in $\mathcal{SHIQ}$ can be reduced to validity of $\mathcal{E}$-safe formulas, and show that $\mathcal{E}$-safe formulas are cardinal.

3.1 Translating to Sorted F-Logic

We first investigate a translation to sorted F-Logic. We augment the translation in Table 3 to ensure that variables are only quantified over the domain of individuals U_i, by replacing each universal quantifier $\forall$ in Table 3 with $\forall_i$ and each existential quantifier $\exists$ with $\exists_i$.

We now show equi-satisfiability of formulas in $\mathcal{L}^P$, and their F-Logic counterparts:

Lemma 1. *Let ϕ be formula in $\mathcal{L}^P$ and let $\mathcal{L}^F$ be the corresponding sorted F-Logic language, then ϕ is satisfied in some interpretation of $\mathcal{L}^P$ if and only if $\delta(\phi)$ is satisfied in some sorted F-structure of $\mathcal{L}^F$.*

Sketch. From any interpretation $\mathcal{I}$ of $\mathcal{L}^P$ such that $\mathcal{I} \models \phi$ one can easily construct a corresponding sorted F-structure $\mathbf{I}$ such that $\mathbf{I} \models_{\mathsf{f}} \delta(\phi)$, and vice versa. □

Theorem 1. *Let $\Phi \subseteq \mathcal{L}^P$ be a set of formulas in the language $\mathcal{L}^P$, $\delta(\Phi) \subseteq \mathcal{L}^F$ be the corresponding F-Logic formulas in a sorted $\mathcal{L}^F$, and let $\phi \in \mathcal{L}^P$ be an arbitrary formula, then:*

$$\Phi \models \phi \textit{ iff } \delta(\Phi) \models_{\mathsf{f}} \delta(\phi)$$

Proof. Follows immediately from Lemma 1 and the fact that checking entailment $\Phi \models \phi$ can be reduced to checking unsatisfiability of $(\bigwedge \Phi) \wedge \neg\phi$. □

3.2 Translating Cardinal Formulas

We now consider the translation function δ of Table 3 in its original form and we consider unsorted F-structures of the form $\mathbf{I} = \langle U, \preceq_U, \in_U, \mathbf{I}_F, \mathbf{I}_P, \mathbf{I}_{\twoheadrightarrow} \rangle$.

It turns out that we lose the correspondence of models in the general case with this augmented definition. Consider, for example, the following formula:

$$\phi = (\forall x, y.x = y) \supset (q(a) \leftrightarrow r(a))$$

The formula ϕ is trivially satisfied in any interpretation with more than one element in the domain, since the antecedent will be trivially false in such an interpretation. If we consider an interpretation with only one element, then the antecedent is true, but the consequent is not necessarily true, because q and r may be interpreted differently. Thus, ϕ is not valid in FOL. Now consider the corresponding F-Logic formula:

$$\delta(\phi) = (\forall x, y.x = y) \supset (a : q \leftrightarrow a : r)$$

As we have seen, the original formula ϕ is not valid in $\mathcal{L}^P$. However, $\delta(\phi)$ is valid in $\mathcal{L}^F$, since q and r must be interpreted as the same class in every F-structure which has exactly one element.

From the example we can see that the translation δ does not work for arbitrary predicate-based ontology languages. There is, however, a class of formulas for which the correspondence does hold with the augmented definition. This is the class of *cardinal* formula as defined in [7].

Definition 2. *Let $\Phi \subseteq \mathcal{L}^P$ be a set of formulas in $\mathcal{L}^P$, let γ denote the number of symbols in $\mathcal{L}^P$, then Φ is* cardinal *with respect to $\mathcal{L}^P$ if the following holds:*

> *If Φ is true in every interpretation $\mathcal{I}$ such that the cardinality of the domain of $\mathcal{I}$ is at least γ, then Φ is true in every interpretation of $\mathcal{L}^P$.*

An interpretation $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ is cardinal if $|\Delta| \geq \gamma$.

With a little abuse of notation we will also write $|\mathcal{I}|$ for $|\Delta|$ in the remainder.

Note that this condition does not hold for the formula ϕ mentioned above, because ϕ is true in every interpretation with a domain of at least 3 elements, but it is not true in every interpretation of $\mathcal{L}^P$. The following definition of cardinality is equivalent to Definition 2:

Corollary 1. *Let ϕ be a formula in $\mathcal{L}^P$, then ϕ is* cardinal *with respect to $\mathcal{L}^P$ if and only if:*

> *If ϕ is true in an interpretation of $\mathcal{L}^P$, then ϕ is true in an interpretation that is cardinal for $\mathcal{L}^P$.*

Proof. Assume ϕ is true in some interpretation $\mathcal{I}$ of $\mathcal{L}^P$, i.e., $\mathcal{I} \models \phi$. The latter is equivalent to $\mathcal{I} \not\models \neg\phi$, thus, by contraposition of Definition 2, there is an interpretation $\mathcal{I}'$ which is cardinal for $\mathcal{L}^P$ such that $\mathcal{I}' \not\models \neg\phi$. The latter is equivalent to $\mathcal{I}' \models \phi$. $\square$

We can now strengthen Lemma 1 and Theorem 1 to the case of unsorted F-Logic:

Lemma 2. *Let ϕ be a formula in $\mathcal{L}^P$. Then*

1. *if $\delta(\phi)$ is satisfied in some F-structure of $\mathcal{L}^F$, then there is an interpretation of $\mathcal{L}^P$ which satisfies ϕ, and*
2. *if ϕ is cardinal and is satisfied in some interpretation of $\mathcal{L}^P$, then there is an F-structure of $\mathcal{L}^F$ which satisfies $\delta(\phi)$.*

Proof. Given a cardinal interpretation $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ of $\mathcal{L}^P$, then $\mathbf{I} = (\mathcal{I})^{FL} = \langle U, \preceq_U, \in_U, \mathbf{I}_F, \mathbf{I}_P, \mathbf{I}_{\twoheadrightarrow} \rangle$ is the corresponding F-Logic structure, which is obtained as follows: (i) $U = \Delta$, (ii) $\forall\ f \in A$: $\mathbf{I}_F(f) = f^{\mathcal{I}}$, (iii) $\forall\ c \in C$: $\mathbf{I}_F(c) = k_c$ for some $k_c \in U$, (iv) $\forall\ r \in R$: $\mathbf{I}_F(r) = k_r$ for some $k_r \in U$, (v) $\forall\, c \in C$ and every individual $k \in \Delta$, if $k \in c^{\mathcal{I}}$ then $k \in_U \mathbf{I}_F(c)$, (vi) $\forall\ c_1, c_2 \in C$: if $c_1^{\mathcal{I}} \subseteq c_2^{\mathcal{I}}$ then $\mathbf{I}_F(c_1) \preceq_U \mathbf{I}_F(c_2)$, (vii) $\forall\ r \in \mathcal{R}$ and $\forall\ k_1, k_2 \in \Delta$, if $\langle k_1, k_2 \rangle \in r^{\mathcal{I}}$ then $k_2 \in \mathbf{I}_{\twoheadrightarrow}(\mathbf{I}_F(r))(k_1)$, and (viii) $\forall\ p \in \mathcal{P}$: $\mathbf{I}_P(p) = p^{\mathcal{I}}$. All k_c and k_r must be mutually disjoint. Since $\mathcal{I}$ is cardinal, obviously there is such an $\mathbf{I}$.

Given an F-structure $\mathbf{I} = \langle U, \preceq_U, \in_U, \mathbf{I}_F, \mathbf{I}_P, \mathbf{I}_{\twoheadrightarrow} \rangle$ for the language $\mathcal{L}^F$, the corresponding FOL interpretation $\mathcal{I} = (\mathbf{I})^{FOL} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ for $\mathcal{L}^P$ is defined as follows: (i) $\Delta = U$, (ii) $\forall\ f \in \mathcal{A}$: $f^{\mathcal{I}} = \mathbf{I}_F(f)$, (iii) $\forall\ c \in \mathcal{C}$: $c^{\mathcal{I}} = \{k \mid k \in_U \mathbf{I}_F(c)$ is true for $k \in U\}$, (iv) $\forall\ r \in \mathcal{R}$: $r^{\mathcal{I}} = \{\langle k_1, k_2 \rangle \mid k_2 \in \mathbf{I}_{\twoheadrightarrow}(\mathbf{I}_F(r))(k_1)$, for $k_1, k_2 \in U\}$, and (v) $\forall\ p \in \mathcal{P}$: $p^{\mathcal{I}} = \mathbf{I}_P(p)$. Obviously, there is such an $\mathcal{I}$.

We now proceed to prove the lemma:

(1) Assume $\mathbf{I} \models_{\mathsf{f}} \delta(\phi)$ for some F-structure $\mathbf{I}$, then it is easy to verify that $\mathcal{I} \models \phi$.

(2) Assume $\mathcal{I} \models \phi$ for some interpretation $\mathcal{I}$ and cardinal formula ϕ. By Corollary 1, we have that there is a cardinal model $\mathcal{I}'$ of ϕ.

Assume $\mathcal{I}', B \models \phi$ for some variable assignment B. Since $\mathcal{I}'$ is cardinal, $\mathbf{I} = (\mathcal{I}')^{FL}$ is defined. To prove the lemma, it is sufficient to show that $\mathbf{I}, B \models_{\mathsf{f}} \delta(\phi)$ (we may use the same variable assignment, because $U = \Delta$). We proceed by induction over the length of the formula ϕ.

Consider $\phi = C(X)$. $\mathcal{I}', B \models \phi$ iff $t^{\mathcal{I}', B} \in C^{\mathcal{I}'}$ iff $t^{\mathbf{I}, B} \in_U \mathbf{I}_F(C)$. The 'only if' direction follows from (v) in the translation above. The 'if' direction follows from the fact that $\mathbf{I}_F(C) \neq k$ for any $k = \mathbf{I}_F(D)$, with $D \neq C$ a concept identifier. Similar for formulas of the form $R(X, Y)$.

Consider $\phi = (t_1 = t_2)$. $\mathcal{I}', B \models \phi$ iff $t_1^{\mathcal{I}', B} = t_2^{\mathcal{I}', B}$ iff $t_1^{\mathcal{I}, B} = t_2^{\mathbf{I}, B}$. The last 'iff' follows trivially from the construction of $\mathbf{I}$.

Consider $\phi = \forall x(\psi)$. $\mathcal{I}', B \models \phi$ iff for every x-variant B' of $B, \mathcal{I}', B' \models \psi$ iff $\mathbf{I}, B' \models \delta(\psi)$. The last 'iff' follows by induction and from the observation that the domains of $\mathcal{I}$ and $\mathbf{I}$ are the same. Similar for $\phi = \exists x(\psi)$. This can be trivially extended to formulas of the forms $\neg\psi$, $\psi_1 \wedge \psi_2$, and $\psi_1 \vee \psi_2$. $\square$

Theorem 2. *Let $\Phi \subseteq \mathcal{L}^P$ be a set of formulas and $\phi \in \mathcal{L}^P$ be a formula, then:*

$$\text{if } \Phi \models \phi \text{ then } \delta(\Phi) \models_{\mathsf{f}} \delta(\phi)$$

If $\neg(\bigwedge \Phi) \vee \phi$ is cardinal, we have additionally:

$$\Phi \models \phi \text{ if } \delta(\Phi) \models_{\mathsf{f}} \delta(\phi)$$

Proof. Follows from Lemma 2 and the observation that checking entailment can be reduced to checking validity of $\neg(\bigwedge \Phi) \vee \phi$. $\square$

Results on cardinal formulas for HiLog can be transferred to F-Logic. From [7] we know that equality-free sentences, as well as negation of Horn clauses with no equality in the antecedent are cardinal. This is, however, not sufficient for many ontology languages which allow the assertion of equality between individuals and maximal number restrictions, such as the Description Logic $\mathcal{SHIQ}$.

We define the class of $\mathcal{E}$-safe formulas ($\mathcal{E}$ stands for "equality") which allow only *safe* uses of equality. With "safe" we mean that the use of the equality does not restrict the size of the domains of the models. The structure of $\mathcal{E}$-safe formulas is similar to the structure of *guarded* formulas [1]. For $\mathcal{E}$-safe formulas, the guard ensures that equality statements range only over part of the domain.

We first define the class of *limited* $\mathcal{E}$-safe formulas, denoted $l\mathcal{ESF}$:

$$l\mathcal{ESF} ::= \quad A \mid \neg A \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \\ \forall \vec{x}(\chi \supset \phi) \mid \exists \vec{x}(\chi \wedge \phi)$$

where A is an atomic formula either of the form $p(\vec{t})$ or $t_1 = t_2$ with t_1, t_2 either both ground or non-ground terms; ϕ, ϕ_1, ϕ_2 are $l\mathcal{E}$-safe formulas, and χ is either an atom of the form $p(\vec{t})$ or a conjunction of atoms of the form $p(\vec{t})$ such that every atom in χ with free variables shares at least one variable with another atom[2]. Finally, every free variable in ϕ must be appear in χ. We now define the class of $\mathcal{E}$-safe formulas, denoted $\mathcal{ESF}$:

$$\mathcal{ESF} ::= \quad \varphi \mid \forall x(\phi) \mid \exists x(\phi) \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2$$

With ψ_1, ψ_2 $\mathcal{E}$-safe formulas, ϕ, φ $l\mathcal{E}$-safe formulas, and x the only free variable in ϕ. As usual, an $\mathcal{E}$-safe sentence is an $\mathcal{E}$-safe formula without free variables.

We consider formulas of the forms $\forall x(x = x \supset \phi)$ and $\exists x(x = x \wedge \phi)$, with ϕ an $l\mathcal{E}$-safe formula with one free variable x, $\mathcal{E}$-safe, because they are equivalent to $\forall x(\phi)$ and $\exists x(\phi)$, respectively. As is usual in guarded logics, we thus assume that formulas $\forall x(\phi)$, $\exists x(\phi)$ are guarded by $x = x$.

Notice that the negation of an $\mathcal{E}$-safe formula is $\mathcal{E}$-safe as well.

Example 1. *The following formulas are $\mathcal{E}$-safe:*
$\forall x.(p(x) \supset q(x))$
$\forall x.(s(x, y) \supset p(x))$
$\exists x, y.p(x) \wedge r(x, y) \wedge x = y$
$\forall x.r(x)$
The following formulas are not $\mathcal{E}$-safe:
$\forall x, y.x = y$
$\forall x.a(x) \wedge a(y) \supset x = y$
$\forall x.x = y \supset p(x, y)$

Many expressive Description Logic languages are $\mathcal{E}$-safe, including $\mathcal{SHIQ}$:

Proposition 1. *Any (negation of a) $\mathcal{SHIQ}$ axiom ϕ can be rewritten to an $\mathcal{E}$-safe formula ϕ' such that ϕ and ϕ' are equivalent, i.e., share the same models.*

Proof. Assume ϕ is the first-order version a $\mathcal{SHIQ}$ axiom (translation of $\mathcal{SHIQ}$ axioms to FOL formulas can be done according to Table 2). In case ϕ is a property or individual axiom, it is trivially $\mathcal{E}$-safe and $\phi' = \phi$.

Say, ϕ is a class axiom of the form $\phi \equiv \forall x(\phi_1 \supset \phi_0)$. Given the form of ϕ and the translation in Table 2, one can transform $\phi_1 \supset \phi_0$ to a conjunction ψ of $l\mathcal{E}$-safe formulas, e.g., removing disjunction from the antecedent induces a splitting of the original formula in a conjunction of formulas, such that $\phi' \equiv \forall x \psi$ is an $\mathcal{E}$-safe formula that is equivalent to ϕ.

As the negation of an $\mathcal{E}$-safe formula is again an $\mathcal{E}$-safe formula we have that the negation of a $\mathcal{SHIQ}$ axiom is $\mathcal{E}$-safe as well. □

Note that $\mathcal{SHOIQ}$ formulas are not $\mathcal{E}$-safe in general, because of the possibility of using nominals. Consider, for example, the $\mathcal{SHOIQ}$ knowledge base $\{\top \sqsubseteq \{a\}\}$. This is equivalent to the first-order sentence $\forall x(x = a)$. Every model of this knowledge base has exactly one element in its domain. This generalizes to any Description Logic with unrestricted use of nominals.

$\mathcal{E}$-safe formulas are a highly expressive class of formulas. In fact, it is easy to see, using a modification of Proposition 1, that $\mathcal{SHIQ}$ knowledge bases extended with Horn formulas can be equivalently translated to sets of $\mathcal{E}$-safe formulas. As entailment in the former is undecidable in general [19], entailment of $\mathcal{E}$-safe formulas is undecidable in general, as well.

We now formulate our main result with respect to cardinal formulas:

Lemma 3. *The following classes of formulas first-order formulas are cardinal:*

1. *Sets of equality-free sentences,*
2. *formulas of the form $\neg S$, where S is a conjunction of Horn clauses without equality in the head,*
3. *the class of $\mathcal{E}$-safe sentences.*

Proof. Cardinality of the first and second class is shown in [7]. We proceed with the proof of cardinality of $\mathcal{E}$-safe formulas.

There are five types of $\mathcal{E}$-safe sentences: (1) $l\mathcal{ESF}$ sentences, (2) universal and (3) existential $\mathcal{E}$-safe sentences and (4) conjunctions and (5) disjunctions of $\mathcal{E}$-safe sentences. Any $l\mathcal{ESF}$ sentence ϕ can be equivalently written as a universal sentence $\forall x(\phi)$. We now proceed to prove cardinality of sentences of the forms (2,3,4,5).

We need the following auxiliary notion. Given an interpretation $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$, $k \in \Delta$ is *unused* in $\mathcal{I}$ if: (a) k does not occur in the domain or the range of a function $f^{\mathcal{I}} : \Delta^n \to \Delta$ for $f \in \mathcal{A}$, and (b) k does not occur in a relation $p^{\mathcal{I}} : \Delta^n$ for $p \in \mathcal{C} \cup \mathcal{R} \cup \mathcal{P}$.

(2) We proceed by induction. Assume $\mathcal{I}^{\geq\gamma} \models \forall x(\phi)$ for every cardinal interpretation $\mathcal{I}^{\geq\gamma}$. We will show that if $\mathcal{I}^{i+1} \models \forall x(\phi)$ for every interpretation $\mathcal{I}^{i+1}$ of cardinality $i + 1$, then $\mathcal{I}^{i} \models \phi$ for every interpretation $\mathcal{I}^{i}$ of cardinality i, with $i \geq 1$. By induction, this guarantees that every interpretation is a model of $\forall x(\phi)$, and thus the formula is cardinal. Let $\mathcal{I}^{i}$ be an interpretation of cardinality i, and let

[2] The atoms must form a "chain", where the shared variables are the links in the chain.

$\mathcal{I}^{i+1}$ be the interpretation obtained from $\mathcal{I}^i$ by adding one unused individual to the domain. By the induction hypothesis, $\mathcal{I}^{i+1} \models \forall x(\phi)$. Thus, for every variable assignment B of $\mathcal{I}^{i+1}$, $\mathcal{I}^{i+1}, B \models \phi$. Since the domain of $\mathcal{I}^i$ is a subset of the domain of $\mathcal{I}^{i+1}$, every variable assignment of $\mathcal{I}^i$ is a variable assignment of $\mathcal{I}^{i+1}$. Thus, for every variable assignment B' of $\mathcal{I}^i$, $\mathcal{I}^{i+1}, B' \models \phi$. We now show by induction over the length of the formula ϕ that if $\mathcal{I}^{i+1}, B' \models \phi$, then $\mathcal{I}^i, B' \models \phi$.

If $\mathcal{I}^{i+1}, B' \models (t_1 = t_2)$, then $t_1^{\mathcal{I}^{i+1},B'} = t_2^{\mathcal{I}^{i+1},B'}$; clearly, $t_1^{\mathcal{I}^{i},B'} = t_1^{\mathcal{I}^{i+1},B'}$ and $t_2^{\mathcal{I}^{i},B'} = t_2^{\mathcal{I}^{i+1},B'}$, and thus $t_1^{\mathcal{I}^{i},B'} = t_2^{\mathcal{I}^{i},B'}$ and $\mathcal{I}^i, B' \models (t_1 = t_2)$.

If $\mathcal{I}^{i+1}, B' \models p(t_1, ..., t_n)$, then $\langle t_1^{\mathcal{I}^{i+1},B'}, ..., t_n^{\mathcal{I}^{i+1},B'}\rangle \in p^{\mathcal{I}^{i+1}}$ and thus $\mathcal{I}^i, B' \models \phi$.

If $\mathcal{I}^{i+1}, B' \models \neg(t_1 = t_2)$ then $t_1^{\mathcal{I}^{i+1},B'} \neq t_2^{\mathcal{I}^{i+1},B'}$, and by the same argument as above, $\mathcal{I}^i, B' \models \neg(t_1 = t_2)$. Similar for $\mathcal{I}^{i+1}, B' \models \neg p(t_1, ..., t_n)$.

If $\mathcal{I}^{i+1}, B' \models \psi_1 \wedge \psi_2$, $\mathcal{I}^i, B' \models \psi_1$ and $\mathcal{I}^i, B' \models \psi_2$, then, clearly, $\mathcal{I}^i, B' \models \psi_1 \wedge \psi_2$. Similar for $\psi_1 \vee \psi_2$.

If $\mathcal{I}^{i+1}, B' \models \exists\vec{x}(\chi \wedge \phi)$, then there is an $\vec{x}$-variant B'' of B' such that $\mathcal{I}^{i+1}, B'' \models \chi \wedge \phi$. Assume B'' assigns a free variable in χ to an unused individual in $\mathcal{I}^{i+1}$, then, clearly, $\mathcal{I}^{i+1}, B'' \not\models \chi$. Therefore, we may assume that B'' is an $\vec{x}$-variant of B' which does not assign any variable to an unused individual, and $\mathcal{I}^{i+1}, B'' \models \chi \wedge \phi$. By induction we have, $\mathcal{I}^i, B'' \models \chi$ and $\mathcal{I}^i, B'' \models \phi$, and thus $\mathcal{I}^i, B' \models \exists\vec{x}(\chi \wedge \phi)$.

If $\mathcal{I}^{i+1}, B' \models \forall\vec{x}(\chi \supset \phi)$, then $\mathcal{I}^{i+1}, B'' \models \chi \supset \psi$ for every $\vec{x}$-variant B'' of B' of $\mathcal{I}^i$ (by the same argument as the outer induction). Clearly, if $\mathcal{I}^{i+1}, B'' \not\models \chi$, then $\mathcal{I}^i, B'' \not\models \chi$, since χ is a conjunction of atomic formulas. By induction we have that if $\mathcal{I}^{i+1}, B'' \models \phi$, then $\mathcal{I}^i, B'' \models \phi$, and thus $\mathcal{I}^{i+1}, B' \models \forall\vec{x}(\chi \supset \phi)$.

(3) If $\mathcal{I} \models \exists x(\phi)$, then there is a variable assignment B such that $\mathcal{I}, B \models \phi$. Let $\mathcal{I}^{\mathsf{c}}$ be a cardinal interpretation obtained from $\mathcal{I}$ by adding a sufficient number of unused individuals to the domain. It is easy to verify using induction over the length of the formula, similar to the induction in (2), that if $\mathcal{I}, B \models \phi$, then $\mathcal{I}^{\mathsf{c}}, B \models \phi$ for ϕ an $l\mathcal{ESF}$ formula (note that B is a variable assignment of $\mathcal{I}^{\mathsf{c}}$, because the domain of $\mathcal{I}^{\mathsf{c}}$ is a superset of that of $\mathcal{I}$). Thus, by Corollary 1, $\exists x(\phi)$ is cardinal.

(4) Assume ψ_1, ψ_2 are cardinal. Now, if every cardinal interpretation $\mathcal{I}$ is a model of $\psi_1 \wedge \psi_2$, then every cardinal interpretation is a model of ψ_1 and ψ_2 and, by cardinality of ψ_1, ψ_2, every interpretation is a model of ψ_1 and ψ_2. Therefore, every interpretation is a model of $\psi_1 \wedge \psi_2$ and thus $\psi_1 \wedge \psi_2$ is cardinal.

(5) Assume ψ_1, ψ_2 are cardinal. If $\mathcal{I} \models \psi_1 \vee \psi_2$ then $\mathcal{I} \models \psi_1$ or $\mathcal{I} \models \psi_2$. Say $\mathcal{I} \models \psi_1$, then, by cardinality of ψ_1 and Corollary 1, there is a cardinal interpretation $\mathcal{I}'$ such that $\mathcal{I}' \models \psi_1$; similar for ψ_2. Thus, there is a cardinal interpretation $\mathcal{I}'$ such that $\mathcal{I}' \models \psi_1 \vee \psi_2$ and thus $\psi_1 \vee \psi_2$ is cardinal. $\square$

The following corollary follows immediately from Theorem 2, Proposition 1 and Lemma 3:

Corollary 2. *Let Φ be a set of $\mathcal{SHIQ}$ axioms and ϕ a $\mathcal{SHIQ}$ axiom, then*

$$\Phi \models \phi \textit{ iff } \delta(\Phi) \models_{\mathsf{f}} \delta(\phi)$$

We conclude this section with the observation that the results of Lemma 3 immediately apply to HiLog, since our definition of cardinality coincides with the definition of cardinality in [7]. The following Corollary follows from Lemma 3 and the results in [7].

Corollary 3. *Let ϕ be an $\mathcal{E}$-safe sentence, then ϕ is valid in HiLog if and only if ϕ is valid in FOL.*

4 F-Logic DLP

Description Logic Programs (DLP) [14] can be seen as the expressive intersection of Description logics and logic programming. The Description Logic $\mathcal{DHL}$ is the Horn logic subset of an expressive Description Logic. We follow here the definition of $\mathcal{DHL}$ given in [11], since it includes a slightly larger subset of $\mathcal{SHOIN}$ (the language underlying OWL DL) than the original definition in [14]. A Description Logic Program (DLP) Π_O is obtained from a $\mathcal{DHL}$ ontology O by rewriting the axioms in the ontology to Horn formulas and interpreting the formulas using the standard minimal Herbrand semantics (see e.g. [20]). By the standard results in Logic Programming, we know that O and Π_O agree on ground entailment.

$\mathcal{DHL}$ descriptions are of the following form, where A is an atomic concept, C, D are general descriptions, and C_L, D_L (resp. C_R, D_R) are descriptions which are allowed on the left-hand (resp. right-hand) side of the inclusion symbol $\sqsubseteq$, R, S are atomic roles, o is an individual symbol:

$$\begin{array}{lll} C, D & \longrightarrow & A \mid C \sqcap D \mid \exists R.\{o\} \\ C_L, D_L & \longrightarrow & C \mid C_L \sqcup D_L \mid \exists R.C_L \mid \geqslant 1 R_L \mid \\ & & \{o_1, \ldots, o_n\} \\ C_R, D_R & \longrightarrow & C \mid \forall R.C_R \end{array}$$

A $\mathcal{DHL}$ ontology consists of axioms:

$C_L \sqsubseteq D_R \mid C \equiv D \mid R \sqsubseteq S \mid R \equiv S \mid R \equiv S^- \mid \mathsf{Trans}(R) \mid \top \sqsubseteq \forall R^-.C_R \mid \top \sqsubseteq \forall R.C_R \mid a \in A \mid \langle a, b\rangle \in R$

There are several proposals for layering F-Logic programming on top of $\mathcal{DHL}$ (e.g. [17, 10, 2, 6]. We show that this layering is justified:

Proposition 2. *Let O be a $\mathcal{DHL}$ ontology and let $\pi(O)$ be the FOL equivalent, with π as defined in Table 2, then, for the F-Logic theory $\delta(\pi(O))$, with δ as in Table 3 holds:*

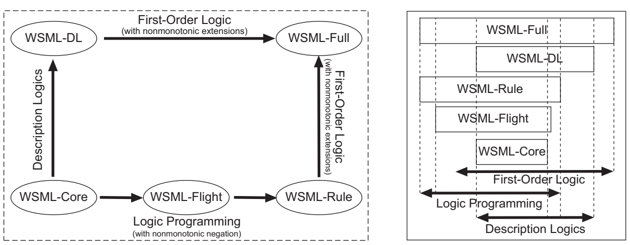
$$O \models \alpha \textit{ iff } \delta(\pi(O)) \models_{\mathsf{f}} \delta(\alpha)$$

with α an equality-free ground atomic formula.

Proof. Equivalence (with respect to entailment, modulo the transformation δ) between $\pi(O)$ and $\delta(\pi(O))$ follows from Theorem 2, Lemma 3 and the fact that $\pi(O)$ is equivalent to a set of Horn formulas without equality in the head. □

5 WSML Layering

Figure 1(a) shows the different variants of the Web Service Modeling Language (WSML) and the relationships between them. These variants differ in logical expressiveness and in the underlying language paradigms.



(a) Language variants (b) Layering

Figure 1. WSML Variants and Layering

WSML-Core is based on by the intersection of the Description Logic $\mathcal{SHIQ}$ and Horn Logic, based on Description Logic Programs [14].

WSML-DL captures the Description Logic $\mathcal{SHIQ}(\mathbf{D})$.

WSML-Flight is based on the Datalog subset of F-Logic programming variant, extended with inequality and (locally) stratified negation under the perfect model semantics [22].

WSML-Rule is based on F-Logic programming, extended with inequality and negation under the Well-Founded semantics [13].

WSML-Full unifies WSML-DL and WSML-Rule under a First-Order umbrella with nonmonotonic extensions. The semantics of WSML-Full is ongoing research.

As shown in Figure 1(b), WSML has two alternative layerings, namely, WSML-Core ⇒ WSML-DL ⇒ WSML-Full and WSML-Core ⇒ WSML-Flight ⇒ WSML-Rule ⇒ WSML-Full. For both layerings, WSML-Core and WSML-Full mark the least and most expressive layers. The two layerings are to a certain extent disjoint in the sense that interoperation in WSML between the Description Logic variant (WSML-DL) on the one hand and the Logic Programming variants (WSML-Flight and WSML-Rule) on the other, is only possible through a common core (WSML-Core) or through a very expressive superset (WSML-Full).

The original WSML specification [9] did not show any semantic properties of this layering. We will first demonstrate the layering WSML-Core ⇒ WSML-DL ⇒ WSML-Full with respect to entailment, and the layering WSML-Core ⇒ WSML-Flight ⇒ WSML-Rule with respect to ground entailment. We cannot demonstrate the layering WSML-Rule ⇒ WSML-Full, because WSML-Full has not been fully specified yet.

For reasons of convenience, clarity and space, we do not consider the WSML syntax in this section, but rather the FOL and F-Logic equivalents, as defined in [9, Chapter 8].

WSML-Core ⇒ WSML-DL A WSML-Core ontology O_{core} consists of the first-order equivalent of a set of $\mathcal{DHL}$ axioms without nominals. O_{core} Core-entails a WSML-Core formula ϕ, denoted $O_{core} \models_{core} \phi$, iff for every first-order model $\mathcal{I}$ of $O_{core}, \mathcal{I} \models \phi$.

A WSML-DL ontology O_{dl} consists of the first-order equivalent of a set of $\mathcal{SHIQ}$ axioms. O_{dl} DL-entails a formula ϕ, denoted $O_{dl} \models_{dl} \phi$, iff for every first-order model $\mathcal{I}$ of $O_{dl}, \mathcal{I} \models \phi$.

Theorem 3. *Given a WSML-Core ontology O_{core}, and a WSML-Core formula ϕ,*

$$O_{core} \models_{core} \phi \textit{ iff } O_{core} \models_{dl} \phi$$

Proof. Follows from the observation that every WSML-Core ontology is a WSML-DL ontology. □

WSML-DL ⇒ WSML-Full We consider, for now, the first-order logic subset of WSML-Full, which we will denote with WSML-FOL.

A WSML-FOL ontology O_{fol} consists of a set of closed first-order F-Logic formulas, as defined in Section 2. We say that a WSML-FOL ontology O_{fol} FOL-entails a formula ϕ, denoted $O_{fol} \models_{fol} \phi$, iff for every F-structure $\mathcal{I}$ which is a model of $O_{fol}, \mathcal{I} \models_{\mathsf{f}} \phi$.

Theorem 4. *Given a WSML-DL ontology O_{dl}, and a WSML-DL formula ϕ,*

$$O_{dl} \models_{dl} \phi \textit{ iff } \{\delta(\psi) \mid \psi \in O_{dl}\} \models_{fol} \delta(\phi)$$

Proof. Follows immediately from Corollary 2. □

WSML-Core $\Rightarrow$ WSML-Flight A WSML-Flight ontology O_{flight} consists of a set O^R_{flight} of F-Logic Datalog rules, extended with locally stratified negation under the perfect model semantics [22] (c.f. [18]), and a set of integrity constraints O^C_{flight}, which are rules without a head.

O_{flight} is consistent if the perfect model M does not violate any of the integrity constraints in O^C_{flight}. An integrity constraint $c \in O^C_{flight}$ is violated in M if the body of c is true in M for some variable substitution θ.

A consistent WSML-Flight ontology O_{flight} Flight-entails a ground atomic formula α, denoted $O_{flight} \models_{flight} \alpha$, iff for every perfect model M of O_{flight}, $M \models \alpha$.

Theorem 5. *Given a WSML-Core ontology O_{core}, and an atomic WSML-Core formula α,*

$$O_{core} \models_{core} \alpha \textit{ iff } \{\delta(\psi) \mid \psi \in O_{core}\} \models_{flight} \delta(\alpha)$$

Proof. Follows immediately from Proposition 2 and the observation that $O^C_{flight} = \emptyset$. $\square$

WSML-Flight $\Rightarrow$ WSML-Rule A WSML-Rule ontology O_{rule} consists of a set O^R_{rule} of F-Logic Horn rules, extended with (un-stratified) negation under the well-founded semantics [13] (c.f. [23]), and a set of integrity constraints O^C_{rule}, which are rules without a head.

O_{rule} is consistent if the well-founded model M of O^R_{rule} does not violate any of the integrity constraints in O^C_{rule}. An integrity constraint $c \in O^C_{rule}$ is violated in M if the body of c is true in M for some variable substitution θ.

We say that a consistent WSML-Rule ontology O_{rule} Rule-entails a ground atomic formula α, denoted $O_{rule} \models_{rule} \alpha$, iff $M \models \alpha$.

Theorem 6. *Given a WSML-Flight ontology O_{flight}, and an atomic WSML-Flight formula α,*

$$O_{flight} \models_{flight} \alpha \textit{ iff } O_{flight} \models_{rule} \alpha$$

Proof. Follows from the fact that O_{flight} is a locally stratified logic program and that for locally stratified logic programs the single (total) well-founded model is also the perfect model of the program [13]. It is easy to see that O_{flight} is a consistent WSML-Flight ontology iff O_{flight} is a consistent WSML-Rule ontology. $\square$

Layering in WRL The Web Rule Language WRL [2] is a proposal for a rule language for the Web, based on WSML. To be more precise, WRL-Core, WRL-Flight, and WRL-Full correspond to WSML-Core, WSML-Flight, and WSML-Rule, respectively. Thus, the layering results obtained in this paper apply immediately to WRL.

6 Related Work

Balaban [4] proposes to use Frame Logic as an underlying framework for description logics and use the flexibility of F-Logic to extend description logics. DFL [5] uses F-Logic to reason about ontologies and rules. The major differences between the approach of Balaban and our approach are: we do not need function symbols if the original language does not use function symbols. We allow arbitrary predicate-based ontology languages, whereas Balaban' translation restricted to Description Logics. Balaban uses a sorted F-Logic, whereas we do not need sorts for a large class of formulas.

F-OWL [25] uses FLORA [24], an F-Logic programming implementation, to reason over OWL. The authors capture the semantics of OWL using entailment rules over RDF triples. It is not clear exactly which part of the semantics of OWL is captured in F-OWL.

Two proposals for extending OWL DL with metamodeling support are presented in [21]. The proposals are based on contextual predicate calculus and HiLog [7]. It was not discussed in [21] whether HiLog-$\mathcal{SHOIQ}$ is a proper extension of $\mathcal{SHOIQ}$ in the sense that a $\mathcal{SHOIQ}$ knowledge base Φ entails an axiom ϕ if and only if Φ HiLog-entails ϕ. We conjecture that by Corollary 3 and the fact that the semantics of HiLog-$\mathcal{SHOIQ}$ is very close to HiLog, HiLog-$\mathcal{SHIQ}$ is a proper extension of $\mathcal{SHIQ}$, but HiLog-$\mathcal{SHOIQ}$ is not a proper extension of $\mathcal{SHOIQ}$; it might be the case that Φ HiLog-entails ϕ, but not Φ entails ϕ. This intuition was confirmed by the author of [21][3].

7 Conclusions

In predicate-based ontology representation languages (e.g. Description Logics), classes are modeled as unary predicates and properties as binary predicates, which are interpreted as sets and as binary relations, respectively. In F-Logic, classes and properties are both first interpreted as objects and then related to sets and relations, respectively.

In this paper we have introduced a translation from predicate-based ontologies to ontologies in F-Logic. We have shown that this translation preserves entailment for large classes of predicate-based ontology languages, including the class of *cardinal* formulas. Intuitively, cardinal formulas do not restrict the size of the domains of the models. We have defined the class of $\mathcal{E}$-safe formulas and shown that $\mathcal{E}$-safe formulas are cardinal. Finally, we have shown that the class of $\mathcal{E}$-safe formulas is a very expressive class of formulas which includes the description logic $\mathcal{SHIQ}$.

We have used the translation to close the open problems of F-Logic extension of Description Logic Programs [14] and WSML language layering [10].

[3] Personal communication.

The results obtained in this paper can be used for F-Logic based reasoning with, and extension of, classes of predicate-based ontology languages. Another application of the results is the use of F-Logic as a vehicle for the extension of RDF, similar to the first-order extensions of RDF described in [8]. This encoding of RDF(S) in F-Logic is future work.

References

[1] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.

[2] J. Angele et al. Web rule language (WRL). W3C Member Submission 09 September 2005.

[3] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.

[4] M. Balaban. The F-logic approach for description languages. *Annals of Mathematics and Artificial Intelligence*, 15(1):19–60, 1995.

[5] M. Balaban and A. Eyal. DFL - a dialog based integration of concept and rule reasoners. *Data & Knowledge Engineering*, 38(3):301–334, 2001.

[6] S. Battle et al. Semantic web services language (SWSL). W3C Member Submission 09 September 2005.

[7] W. Chen, M. Kifer, and D. S. Warren. HILOG: A foundation for higher-order logic programming. *Journal of Logic Programming*, 15(3):187–230, 1993.

[8] J. de Bruijn, E. Franconi, and S. Tessaris. Logical reconstruction of normative RDF. In *OWLED-2005*, Galway, Ireland, November 2005.

[9] J. de Bruijn, editor. The web service modeling language WSML. WSML Final Draft D16.1v0.21, WSML, 2005.

[10] J. de Bruijn, H. Lausen, A. Polleres, and D. Fensel. The web service modeling language: An overview. In *ESWC2006*, Budva, Montenegro, June 2006.

[11] J. de Bruijn, A. Polleres, R. Lara, and D. Fensel. OWL^{-}. Final draft d20.1v0.2, WSML, 2004.

[12] M. Dean and G. Schreiber, editors. *OWL Web Ontology Language Reference*. W3C Recommendation 10 February 2004.

[13] A. V. Gelder, K. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, 1991.

[14] B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: Combining logic programs with description logic. In *WWW2003*, Budapest, Hungary, 2003.

[15] P. Hayes. RDF semantics. W3C Recommendation 10 February 2004.

[16] S. Khoshafian and G. Copeland. Object identity. In *OOPSLA'86*, pages 406–416, 1986.

[17] M. Kifer. Rules and ontologies in f-logic. In *Reasoning Web, First International Summer School, Tutorial Lectures*, pages 22–34, Msida, Malta, July 2005.

[18] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *JACM*, 42(4):741–843, 1995.

[19] A. Y. Levy and M.-C. Rousset. Combining horn rules and description logics in CARIN. *Artificial Intelligence*, 104:165 – 209, 1998.

[20] J. W. Lloyd. *Foundations of Logic Programming (2nd edition)*. Springer-Verlag, 1987.

[21] B. Motik. On the properties of metamodeling in OWL. In *ISWC2005*, pages 548–562, 2005.

[22] T. C. Przymusinski. On the semantics of stratified deductive databases. In *Proceedings of the Workshop on the Foundations of Deductive Databases and Logic Programming*, pages 433–443, Washington, DC, USA, 1986.

[23] G. Yang and M. Kifer. Well-founded optimism: Inheritance in frame-based knowledge bases. In *ODBASE2002*, Irvine, California, 2002.

[24] G. Yang, M. Kifer, and C. Zhao. FLORA-2: A rule-based knowledge representation and inference infrastructure for the semantic web. In *ODBASE2003))*, Catania, Sicily, Italy, 2003.

[25] Y. Zou, T. Finin, and H. Chen. F-OWL: an inference engine for the semantic web. In *FAABS2004*, Greenbelt, USA, 2004.x